ARCHITECTURE AND DESIGN OF A CMOS IC FOR PACKET SWITCHING MULTI-GIGABIT DATA STREAMS

by

Jeremy Ekman

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering

Winter 2005

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2005		**00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Architecture and Design of a CMOS IC for Packet Switching Multi-Gigabit Data Streams	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Delaware,Department of Electrical and Computer Engineering ,Newark,DE,19716**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**174**	

ARCHITECTURE AND DESIGN OF A CMOS IC FOR PACKET SWITCHING MULTI-GIGABIT DATA STREAMS

by

Jeremy Ekman

Approved: _____
Gonzalo R. Arce, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____
Eric W. Kaler, Ph.D.
Dean of the College of Engineering

Approved: _____
Conrado M. Gempesaw II, Ph.D.
Vice Provost for Academic and International Programs

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Fouad E. Kiamilev, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Michael W. Haney, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Charles S. Ih, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Richard G. Rozier, Ph.D.
Member of dissertation committee

ACKNOWLEDGMENTS

The work presented here is a part of a very large research program to develop and demonstrate optical inter-chip communication, which was made possible by the support of DARPA funding. Accordingly, my work has been carried out in close collaboration with many individuals at several organizations. I would specifically like to acknowledge some people with whom I collaborated most closely on the technical effort. These include Michael McFadden, Predrag Milojkovic, and Marc Christensen from Applied Photonics, Kevin Driscoll, Yue Liu, Brian Vanvoorst and Jim Nohava from Honeywell, Gregg Fokken, Jim Kruchowski, and Jim Bublitz from the Mayo Foundation. I would also like to acknowledge Premanand Chandramani, Xiaoqing Wang, Ping Gui, Mayra Sarmiento, Joshua Kramer, Yongrong Zuo, and James Curtis at the University of Delaware who collaborated with me on various aspects of this program.

I would like to thank the members of my committee, Dr. Michael Haney, Dr. Richard Rozier, and Dr. Charles Ih for their help and support. I would especially like to thank my committee chairperson and advisor throughout my graduate career, Dr. Fouad Kiamilev, whose support and guidance has been invaluable. Finally, I would like to thank my family and especially my wife for encouraging me throughout my years as a student.

TABLE OF CONTENTS

LIST OF TABLES ix
LIST OF FIGURES x
ABSTRACT xiii

Chapter

1 INTRODUCTION 1

1.1 Motivation and Overview 1
1.2 Organization of Dissertation 4

2 BACKGROUND OF SWITCH ARCHITECTURES 6

2.1 Physical Architecture 6
2.2 Switching Schemes 7

2.2.1 Circuit switch 8
2.2.2 Packet Switch 9

2.3 Data Buffering and Queuing 15

2.3.1 Buffered Crossbar 16
2.3.2 Shared Memory 16
2.3.3 Output Queuing 17
2.3.4 Input Queuing 18
2.3.5 Combined Input/Output Queuing 19

2.4 Routing Methods 20
2.5 Scheduling 21
2.6 Fabric Organization and Scaling 22

2.6.1 Switching Elements 23
2.6.2 Scaling Switch Fabric 24

3 VIVACE SWITCH ARCHITECTURE 26

3.1 VIVACE Switch Architecture Introduction and Performance 26

3.2 Detailed Architecture and Functionality 29

3.2.1 Overall Architecture 30
3.2.2 VIVACE Network Protocol 33

3.2.2.1 Features 36
3.2.2.2 Data Flow 36
3.2.2.3 Control Set 37

3.2.3 Inward Logic 46

3.2.3.1 Inward Logic Block Diagram 46
3.2.3.2 Inward Logic Implementation 49
3.2.3.3 Inward Logic Contention 52

3.2.4 Outward Logic 58

3.2.4.1 Outward Logic Block Diagram 58
3.2.4.2 Outward Logic Implementation 62
3.2.4.3 Outward Logic Contention 63

3.2.5 Testability, fault tolerance, and configuration 65

3.3 Comparison with Conventional Switch Implementations 69

4 FSOI BACKGROUND AND USE IN THE VIVACE SWITCH 71

4.1 Optoelectronic Components 71

4.1.1 VCSELS 72
4.1.2 Photodetectors 73

4.2 Interface Electronics 74

4.2.1 VCSEL Drivers 74
4.2.2 Receivers 76

4.3 Design considerations 76

4.3.1 Physical design and integration 77
4.3.2 Architectural effect 79
4.3.3 Allocation of devices 81

5 HARDWARE DEMONSTRATION SYSTEM 83

5.1 ASIC development 83

5.1.1 Test ASIC 85

5.1.1.1 VCSEL Driver 86
5.1.1.2 Photodetector Receiver 90
5.1.1.3 Electrical Input/Output Buffers 92
5.1.1.4 Implementation 94
5.1.1.5 Test Results 98
5.1.1.6 Summary 104

5.1.2 Transceiver ASIC 105

5.1.2.1 Functionality 105
5.1.2.2 Cells 106
5.1.2.3 Architecture 108
5.1.2.4 Interface 111
5.1.2.5 Implementation 112
5.1.2.6 Test Results 116
5.1.2.7 Summary 118

5.2 System Description 119

5.2.1 MCM 120
5.2.2 Motherboard 122

5.2.2.1 Functionality 122
5.2.2.2 Architecture and Components 123
5.2.2.3 Implementation 126

5.2.3 Network Interface 128
5.2.4 Optics 130
5.2.5 System Assembly 131

5.3 System Test-bed Evaluation 139

5.3.1 Incremental Testing 140
5.3.2 Completed System Evaluation 147

6 CONCLUSION 151

BIBLIOGRAPHY 155

LIST OF TABLES

Table 1. Pre-defined Out-of-Band Characters 35

Table 2. Primary Data and Control Words 37

Table 3. Steps in Resolving Outward Logic FIFO Filling 41

Table 4. Steps in Resolving VONIC FIFO Filling 44

Table 5. Switch Inward Logic Ports 50

Table 6. Sources of Inward Logic Control Contention 53

Table 7. Switch Outward Logic Ports 63

Table 8. Sources of Outward Logic Control Contention 64

Table 9. Switch Core Externally Accessible Registers 68

Table 10. Test ASIC Verification Plan 97

Table 11. Power Pad to Internal Circuit Allocation 112

Table 12. Transceiver ASIC Pad List 115

Table 13. Full-System Optical Link Summary 120

Table 14. VCSEL Yield 142

Table 15. Results of Detector Testing Using Fiber-Optic Input 145

LIST OF FIGURES

Figure 1. Dataflow through switch (one path, one direction shown) 27

Figure 2. Switch ASIC top-level block diagram..... 31

Figure 3. Optical connectivity diagram for nine chips 33

Figure 4. Protocol naming convention by network location..... 34

Figure 5. Handling of VONIC-generated Throttles..... 42

Figure 6. Handling of Switch OL-generated Throttles 45

Figure 7. Inward Logic block diagram 47

Figure 8. Hierarchy of Inward Logic VHDL code 50

Figure 9. Outward Logic block diagram..... 59

Figure 10. Hierarchy of Outward Logic VHDL code..... 62

Figure 11. Optical link fault recovery strategy..... 66

Figure 12. General VCSEL Driver cell..... 75

Figure 13. General Receiver architecture with TIA and buffer stages 76

Figure 14. VCSEL driver circuit topology 88

Figure 15. Digital to Analog Converter (DAC) circuit topology 89

Figure 16. Transmitter DAC simulation 90

Figure 17. Photodetector receiver circuit topology 91

Figure 18. Annotated photograph of the Test ASIC die 95

Figure 19. VCSEL drive cell 96

Figure 20. Receiver cell 96

Figure 21. Test data from probe testing the bias voltage generation cells used in the LVDS output driver, the transmitter, and the receiver 99

Figure 22. Common-mode test of post-amplifier stage of the receiver circuit. Inputs are tied together and swept for different values of the bias voltage vb 99

Figure 23. Differential-mode test of post-amplifier stage of the receiver circuit. Inputs are driven separately 100

Figure 24. Test data from electrical characterization of the Test ASIC LVDS Output driver 100

Figure 25. Test data from electrical characterization of the Test ASIC VCSEL driver 101

Figure 26. Test data from electrical characterization of the Test ASIC VCSEL driver 101

Figure 27. Test and simulation data showing DC characteristic of one of the feedback transistors within the receiver pre-amplifier 102

Figure 28. Chip-on-board test setup 104

Figure 29. View of a single cluster showing optoelectronic devices and flip-chip pads 109

Figure 30. Cluster architecture for Transceiver ASIC 111

Figure 31. Composite microphotograph of 7.825 x 7.825 mm VIVACE Transceiver ASIC die (shown wirebonded to mechanical MCM without optoelectronic device array) 113

Figure 32. Plot of VCSEL driver uniformity test data 118

Figure 33. Fully assembled MCM and close-up of single SPA 121

Figure 34. Interposer LGA connector 121

Figure 35. Architecture of Motherboard 126

Figure 36. VIVACE Motherboard substrate showing sites for FPGA and MCM attachment 128

Figure 37. VIVACE Optical Network Interface Card (VONIC). Board dimensions are 25.7 cm x 10.7 cm x 157 mm 129

Figure 38. MCM to Motherboard assembly 133

Figure 39. View of optical alignment 135

Figure 40. Sample page from test bed control spreadsheet 138

Figure 41. Snapshot of bit error checking application 139

Figure 42. MCM-based test of VCSEL yield 141

Figure 43. DC VCSEL test illustration 142

Figure 44. VCSEL light output characterization 143

Figure 45. VCSEL light output characterization 144

Figure 46. Fiber-based detector test 145

Figure 47. Analog output for active alignment 146

Figure 48. Light beams of aligned system 147

Figure 49. Photograph and close-up of the fully assembled system 149

ABSTRACT

Communication requirements in high-performance computing systems continue to increase as the processing nodes within these systems grow in capacity. The work described here looks to future solutions to increasing network bandwidth while maintaining scalability within physically constrained systems by using free-space optical links to implement high-density chip-to-chip interconnection. Such links have advantages over their electrical counterparts in their ability to provide reliable, high-performance connectivity within areas of dense signal routing.

In order to address scaled interconnection bandwidth requirements within switched networks, a system design is presented that consists of a custom switch design that uses wide free-space optical channels between multiple integrated circuits on a multi-chip module to form a scalable switch fabric. In order to show the feasibility of such a system, a hardware demonstration based on custom electronics was built and tested. This included a silicon-CMOS chip that was hybridized with a monolithically integrated array of vertical-cavity surface-emitting lasers and photodetectors that implemented the interconnectivity utilized by the switch design. The resulting hardware demonstrated simultaneous optical communication between seven hybrid chips and is unique in the large scale use of free-space optical interconnects based on this technology. The switch architecture will be presented along with the hardware implementation and system test results.

Chapter 1

INTRODUCTION

1.1 Motivation and Overview

As the demand for higher computing capacity continues to grow, the trend in digital systems is toward larger designs that operate at increasingly higher clock rates. This has resulted in greater demands on the interconnection networks that link them and a need for higher bandwidth inter- and intra-system communication. Two architecture features have become more prevalent in interconnection networks and digital systems. Higher bandwidth links are increasingly used closer and closer to the end-user and switched networks are replacing bus architectures. This is evidenced by the continual emergence of ever increasing speeds of switched Ethernet and the re-architecting of personal computers and workstations to replace legacy bus-based interconnect with high-speed switched connections. At the same time, there is growing emphasis on reduced physical size of computing equipment from both consumer demand and infrastructure constraints. In order to meet the challenges of these higher data rate requirements, new technologies and architectures need to be explored.

The work presented in this dissertation is centered on a program sponsored by the Defense Advanced Research Projects Agency (DARPA) called VIVACE. The aim of the solicitation under which this program received funding was the development and demonstration of photonics systems and associated packaging

technologies that demonstrate the benefits of parallel optical interconnect for “in-the-box” optical communication.

VIVACE, which means vivacious or brisk and spirited in Italian, stands for “VCSEL-based Interconnects in VLSI Architectures for Computational Enhancement”. The goal of the VIVACE program was to develop hardware and software to realize the use of free-space optical interconnect technology and parallel-data fiber optic links to enable high-performance switched Local Area Networks (LANs). As an application demonstration, a custom network would connect multiple computer workstations through a switch module, which uses free-space optical links for internal data routing, in order to accelerate a distributed computation. An important part of this was to demonstrate high bisection bandwidth within a multi-chip module by using wide parallel optical links between chips. A team of industry and university partners collaborated to work toward this goal. The work carried out by this multi-disciplinary team included the development of custom optics, electronics, optoelectronic devices, communication protocols and software as well as modeling, packaging development and creation of integration methodologies.

Central to the VIVACE program was the development of hybrid integrated circuits that combined Silicon-CMOS with large optoelectronic device arrays. Packet switch functionality and protocol specific data handling, implemented as digital logic, was to be coupled with analog interface circuitry within the silicon design. Monolithically integrated VCSEL and photodetector arrays combined with the silicon would form a unit from which a larger high-bandwidth switch could be constructed.

Supporting this switch development, the other major electronics hardware components of the VIVACE system were a multi-chip module onto which hybrid ICs would be placed, a printed circuit board to serve as an interface to the MCM, and a network interface card. The network interface card used parallel-data fiber-interface modules to bring data optically in and out of host computers. On the software side, a stressing military application was parallelized to run on multiple compute nodes. Custom internal network protocols were developed to make the most efficient use of the VIVACE hardware.

The inclusion of free-space optical links in this system affords a much richer interconnection fabric than could be realized with an all-electrical implementation. Optical crosstalk immunity, high density of micro lasers and photodetectors, and two-dimensional integration of devices are features of free-space optical links that are exploited within this system to provide high-speed interconnection between the digital logic blocks. These features not only aid in the performance of the interconnect, but also allow for greater design scalability and reduced physical size.

Leading up to the construction of this packet switch, have been several ASIC and system development efforts which have served as stepping-stones. A key element in the progression toward the current system has been a smaller, free-space optically interconnected circuit switch based on similar optoelectronic device arrays and MCM assembly under a predecessor program called FAST-Net [1]. Another important milestone in the development of the electronics for this system has been the successful fabrication and characterization of test-chips which have served to

demonstrate the operation of new circuits used to interface with optoelectronic devices.

The design of the packet-switch ASIC presented here is based on quarter-micron CMOS technology. It implements the logic for a single switch port and includes the analog circuits to realize a forty-four bit path to all other switch chips on a single MCM. The use of wide optical data links between switch ports allows for high-bandwidth and low latency communication inside the switch while also permitting both full connectivity between ports and scalability in operating speed and number of ports.

The author has contributed in all aspects of the electronics hardware design for the VIVACE program. The work specific to the development and implementation of silicon CMOS integrated circuits as well as the system-level integration and demonstration of these ICs will be presented here.

1.2 Organization of Dissertation

This dissertation is organized to provide background information followed by implementation details and test results. Chapter 2 looks at switch architectures and characteristics. These characteristics motivate the design of a custom switch architecture, which is covered in Chapter 3. The network and switch protocol are described along with the motivation for design decisions. The implementation details are covered, giving examples of how problematic situations that decrease the switch performance are avoided. Chapter 4 builds on the switch architecture by describing how the design is integrated into a free-space optically interconnected system of chips on a multi-chip module. General FSOI technology information and information on the specific optoelectronic devices used in this work is given. This is followed in Chapter

5 with a detailed discussion of the hardware development that led to a final demonstration of a functional free-space optically interconnected system. Circuit and ASIC development and test results are combined with a description of the other relevant system components. This is followed by an explanation of the system assembly process and test results achieved throughout the process. Chapter 6 summarizes the progress made here and discusses extending this work in future systems.

Chapter 2

BACKGROUND OF SWITCH ARCHITECTURES

In a survey of literature on switches and switching networks it was found that the term “switch architecture” is commonly used to mean many different things. The intent in this chapter is to describe existing switch architectures in somewhat general terms rather than to provide a detailed review of the intricacies of the numerous switches that have been built or proposed. As such, various aspects of switch architecture will be presented here in an effort to provide a means of classification.

2.1 Physical Architecture

Large data communications switches are generally constructed to operate as rack-mounted systems and consist of many pieces. They typically consist of “line cards”, “fabric cards”, and backplanes. Line cards are associated with a specific port or ports on the switch and the fabric cards provide the switching functionality to interconnect line cards. A switch chassis is often made up of many line cards (14 – 32) and two to four fabric cards. Each fabric card, in turn, has one or more switch fabric ASIC on it that supports many ports [2].

Line cards serve to implement the ports on the switch and provide data conditioning prior to sending data into the switch fabric. Each line card may implement one or more logical and physical port. Line cards generally contain framers, network processors, and traffic managers. Traffic managers work alongside

network processors to take the framed data packets from in input framer to provide packet classification, policing, traffic shaping, and queuing and scheduling. These functions allow the switch to support and enforce different service levels for different customers or classes of packets and to maintain higher throughput by appropriate congestion avoidance [2].

Line cards connect to fabric cards over a backplane. This connection generally takes the form of high-speed serial electrical links and is more recently done according to standards such as Common Switch Interface (CSIX) level 1 and System Packet Interface (SPI) level 3 or 4 [3][4]. The fabric card contains the actual switching functionality, connecting the multiple ports of all of the line cards within the chassis. This may be on one or more switch ASICs. For the case of multiple fabric cards or multiple fabric chips on a card, packets may be distributed or sent inline. The switch fabric is characterized by its bandwidth, latency, jitter, and availability [5].

A physically smaller implementation for low port-count switches is also common. In this so-called “pizza-box” implementation, the line card components and the switch fabric are contained on a single card. This implementation is more common in end-user switches and routers.

2.2 Switching Schemes

One of the traditional architectural classifications of switches is the delineation of a switch as a circuit switch or a packet switch. While there are many definitions of each of these switch types, the fundamental difference is in how the switch is controlled to yield a link between two ports. In a circuit switch, this link is created in advance and kept as a dedicated resource between the two ports until their *call* or communication session is over at which time the link is released or “torn

down". Due to the semi-permanent link, or circuit, which is created, networks based on circuit switching are said to be connection-oriented. In a packet switch, a communication session is divided up into discrete units, called packets, and each individual packet is routed through the switch based on information within the packet header. In a packet switch, no perpetual connection is established between the two ports that are communicating, and as such, a network based on packet switching is often called connectionless.

2.2.1 Circuit switch

Circuit switching architectures were developed and implemented prior to packet switches. One of the most prominent examples of connection-oriented networks using circuit switching is the public telephone network. The first data communications networks were also based on circuit switching, but are now almost entirely implemented using packet switches.

The dedicated link through a circuit switch during a connection and the advance routing setup impacts the type of traffic it is suited to carry. The first telephone networks were inherently circuit switched because an operator had to physically complete a circuit between two callers. While telephone switches have become automated and now carry digital signals rather than analog, the circuit switching architecture is still beneficial for voice traffic and, as such, the worldwide public switched telephone network (PSTN) is still based on circuit switches [9]. The primary benefit is that the latency through the circuit switched network is fixed, resulting in better real-time quality. Additionally, since telephone calls are generally long compared to data communications, the time needed to setup the circuit becomes insignificant. It is likely that telephone carriers will eventually completely phase out

circuit switches in favor of packet switches as end-to-end real-time support becomes more robust and the bandwidth and management advantages outweigh the considerable cost of conversion. In fact, one U.S. company, Sprint Corporation, has begun this process [6].

2.2.2 Packet Switch

The analogy is often given that a packet switching is similar to the delivery method used by the post office whereby each letter or package is routed independently by a non-predetermined path based on the address on the outside of the package. The traditional definition of a packet switching is a communication paradigm where messages are divided into smaller pieces, called packets, each with its own header which is sufficient for the packet to be independently routed through the network [7][8][9]. Variations on this basic idea of packet switching exist including sub-classes such as store-and-forward, wormhole routing, and virtual cut-through. Further characteristics of packet switches and its variants will be presented in more detail following some background.

Two individuals independently developed the idea of electronic packet switching in data communications systems in the 1960's. Paul Baran of Rand Corporation published a set of studies in 1964, which included the concept of packetized data communication networks that he called "distributed adaptive message block switching". At about the same time, Donald Davies, a researcher at the National Physical Laboratory in the United Kingdom, was working on a new communications system and is credited with coining the term "packet switch" in 1967. The need for a new switching paradigm came from the difference in data communications from traditional voice communications. Most notably,

communications between computer nodes on a network tend to be “bursty” having periods of intense communication interleaved with periods of no, or almost no, communication. This type of traffic leads to great inefficiency in a connection-oriented network due to dedicated circuits being kept during the periods of no communication. Packet switching alleviates this inefficiency by only allocating routing resources between two nodes when there is data being sent.

The most significant step in moving from circuit-switched data networks toward the ubiquitous use of packet switches in modern data networks was made in the late 1960’s when the emerging ARPANET adopted packet switching [10]. This was a network sponsored by the United States’ Advanced Research Projects Agency under the direction of Lawrence Roberts and Robert Tayler and is hailed as not only the first major demonstration of a packet switching network but also the forerunner to the modern Internet.

Data messages in a packet switching network are segmented into one more packets prior to transmission. In general, packets making up a message follow the most expedient path through the network and do not necessarily all traverse the same route. The advantage of this is that the message is delivered as rapidly as possible over available network resources. A potential drawback, however, is that packets do not always arrive at the destination in the same order as they were sent. In order to overcome this, the message must be reassembled from the individual packets received at the destination. In contrast to a circuit switch, which guarantees in-order delivery of data by virtue of its architecture, a packet switch can lead to a problem of “jitter” within real-time communications (e.g. voice or video traffic). Although advances in computer and networking technology are beginning to mitigate this

problem, it is one of the reasons that telecommunication networks have remained connection oriented. Another hazard in packet switched networks (and not in their circuit switched counterparts) is the potential for data loss. This can occur because without pre-negotiated and allocated routing, there is the potential for data contention within the network. This is generally mitigated by the inclusion of data storage within the network, but for arbitrarily large networks the amount of storage required to completely eliminate contention is unbounded. In data networks this is not a large problem because the cost savings achieved by using packet switching far outweighs the cost of implementing end-to-end packet loss detection and retransmission algorithms.

A store-and-forward network is a type of packet switching network in which packets are stored in full before being forwarded onto the next link toward their final destination. It is noted that some authors classify a packet switch as type of store-and-forward switch in contrast to how it is being presented here. However, it is felt that this classification is less suited to the discussion at hand and that a packet switch is more general than a store-and-forward switch.

An important characteristic of any switch is its latency. Since store-and-forward switches store the entire packet prior to transmitting any part of it, the latency is a direct function of the packet length. Since the packet is stored at each intermediate routing point, large store-and-forward networks can incur high latency. One application in which the store-and-forward architecture is especially useful is multi-rate Ethernet switches. Due to the potential disparity between the line rates of different ports (i.e. 10 Mbps, 100 Mbps, 1000 Mbps), the storage of packets allows different generations of Ethernet network cards to communicate without other data

conversion hardware [11]. As such, many commercial desktop and managed Ethernet switches make use of the store-and-forward architecture [12][13].

A routing method for packet switching called *virtual cut-through* was proposed by Kermani and Kleinrock in [14]. In virtual cut-through routing, a message is only stored if the next link required by it is already in use. As a result, for packets which do not encounter blockages in the routing network the latency is improved compared to store-and-forward switching. This method reduces to store-and-forward, however, in the worst case that a given packet is blocked prior to being transmitted for each link it traverses. A disadvantage to virtual cut-through is that since the packets are not stored, data checking is not done within the switch which can lead to a loss in throughput due to the transmission of invalid packets and requires end-to-end checking and retransmission to be handled outside of the switch fabric [11].

Wormhole routing was first proposed in 1986 by Dally and Seitz and is a technique whereby packets traveling through the network are forwarded to the destination or next switching node as soon as the packet header has been examined. [15]. Under this technique, packets are further divided into smaller units called flow-control units, or *flits*. Header flits are followed by data flits. Unlike the separate packets, the flits making up a packet always follow the same route through the network in a pipelined fashion and flits from different packets are not interleaved. Wormhole routing is a modification of virtual cut-through routing. The primary difference between the two is that in wormhole routing, if the header flit is blocked, it and all associated data flits following it are buffered at their current location rather than accumulating at the point of the blockage. This is advantageous because only enough buffer space is needed for the flit as opposed to the whole packet. As in

virtual cut-through routing, wormhole routing has an advantage over store-and-forward switching in that the dependence of latency on the number of links that a message must traverse on its way the destination is largely removed.

A network using wormhole routing is not, in general, non-blocking as evidenced by the need for flit buffers. Furthermore, there is a possibility of deadlock –the network condition where no message can advance because it is blocked by other messages, which also cannot advance. There are, however, methods of mitigating blocking and increasing the network performance. These include the use of queues, internal speedup, and virtual channels [16]. Virtual channels are used to map multiple channels onto a single physical channel and are implemented using parallel buffers for flit storage.

Wormhole routing is prevalent in high-performance computer systems [17]. It has been used in computing systems based on direct networks (near neighbor communication) including the N-Cube Company nCUBE-2/3, Intel Paragon and iPSC, MIT J-Machine [18], Stanford DASH [19], and Cray T3D [20]. Wormhole routing has also been used in computing systems with indirect (or switched) networks such as the Connection Machines CM-5 [21], and IBM SP1 and SP2 [22]. An overview of wormhole routing in some of these systems is also given in [23].

It is illustrative to examine the latency introduced by store-and-forward, virtual cut-through, and wormhole routing networks since it is of great importance especially in large multi-computer systems as well as to compare the three methods. As given in [24], The latency for a store-and-forward network is given by

$$\text{Latency}_{SAF} = (L/B)*D \qquad \text{Eq. 1}$$

where L is the length of the packet, B is the Bandwidth of the channel, and D is the number of links which must be traversed between the source and destination. For virtual cut-through the latency is

$$\text{Latency}_{VC} = (L_h/B)*D + L/B \qquad \text{Eq. 2}$$

Where the new term, L_h, is the length of the header. Finally, for wormhole routing, the latency is given by

$$\text{Latency}_{WR} = (L_f/B)*D + L/B \qquad \text{Eq. 3}$$

and L_f is the length of the flit. For virtual cut-through and wormhole routing, the latency is dominated by the term L/B if $L_h << L$, and $L_f << L$ respectively. Since this is generally the case, the desired independence on the distance from source to destination is observed. It can also be seen that virtual cut-through and wormhole routing bring the latency to a value similar to (or lower than) that obtained in a circuit switched network by considering its latency

$$\text{Latency}_{CS} = (L_c/B)*D + L/B \qquad \text{Eq. 4}$$

where L_c is the length of time needed to set up the connection. Some authors (for example [16]) consider wormhole routing to be a kind of circuit switching or a cross between circuit and packet switching.

From the equations above, it is clear that packetization impacts the overall throughput of the switch due to the overhead of the header information and just as circuit switching of long messages reduces the impact of call setup time, packet switching of long packets lessens the effect of this overhead. However, increasing packet size also means that the inefficiencies of circuit switching in data communications take effect due to wasted bandwidth resulting from the loss of fine-grained switching capability. There are two methods to handle this architecture trade-

off. One is to select a fixed-size unit of data which balances the impact of header overhead with unused bandwidth based on the application. Such packets, which have a fixed size, are often called *cells* such as those in Asynchronous Transfer Mode (ATM) switching which uses a 53-byte cell. Many switch fabric chipsets use cell-based data transmission. Using a fixed size cell as the unit of transmission can lead to better control of bandwidth allocation in the network (i.e. ability to provide different quality of service to different customers accessing the switching network). The other option is not to enforce a packet length allowing small data units to be sent through the network without wasted bandwidth due to a large sized packet and allowing large packets to be transmitted without excessive overhead due to segmenting into many small packets. Switch chipset vendors often refer to this variable data unit size as a packet and there are also many commercial offerings in this area. Extending the idea of (variable sized) packets such that a complete message is contained within only one packet is called message switching. Notwithstanding these differences in data format, it is typical in switches, that packets of whatever length are first transformed into a fixed size cell, which traverses the switch and is then converted back into its original format.

2.3 Data Buffering and Queuing

Packet switches do not maintain a dedicated link between ports during a communication session. Therefore link contention or blocking is possible. In order to resolve contention either within the switch or at its output ports data storage is required. This is implemented within the switch fabric, at the input ports, at the output ports, or at some combination of these, using random access memory. Architectures with memory added inside the switch fabric may be classified as shared memory

switches or buffered crossbars. Adding memory queues outside the switch fabric, at the ports, may be classified as output queuing, input queuing, or combined input output queuing. The overall switch may actually have memory in multiple places regardless of the memory architecture and be classified based on what queues are being considered by the arbitration algorithm.

2.3.1 Buffered Crossbar

The distinguishing feature of buffered crossbar architecture is the presence of memory within the switch fabric and the scheduling method. In addition to the fabric memory, there are queues at the input and the output ports. The scheduling of when each of these queues is serviced is done in a distributed manor. This is advantageous because de-centralized scheduling is simpler to implement in hardware and requires less global communication. The trade-off is in the realization of quality-of-service features. Without global knowledge of the current traffic within the switch, it is difficult to schedule traffic according to priority, for instance.

2.3.2 Shared Memory

The shared memory architecture is similar to a time domain switch or shared medium switch except that the shared resource is the link to the memory and the memory itself. This architecture has a single central memory into which all incoming data is immediately queued. Data is taken out of this dual-port memory concurrently according to the scheduling algorithm and placed on the output ports. Because the queues are logically located at the output ports, this is a type of output queuing. For a shared memory switch with N input and output ports, the bandwidth of the memory must be:

$$BW_{mem\text{-}SM} = 2*N*R \quad \text{Eq. 5}$$

where R is the line rate of each port and the factor of two comes from simultaneous read and write. This dependence of the required memory bandwidth on the number of ports, which results in poor scaling, is the primary disadvantage of this architecture. For small switches that have a high line rate serial inputs and outputs, performing serial to parallel conversion at the inputs and parallel to serial conversion at the outputs can reduce the memory bottleneck. The benefit of output queuing is its efficiency in handling bursty traffic. The buffer space of a shared memory switch can be dynamically allocated to output ports allowing more memory to be temporarily used by the ports that most need it. Therefore, for the same amount of memory space, the shared memory switch can experience fewer dropped packets than an output queued switch with a dedicated memory for each output port [25]. This benefit and the simple nature of the architecture have led to its use in many of the early commercial packet switch chip sets as well as some current ones [26].

2.3.3 Output Queuing

Switches that resolve output port contention by providing buffers at the switch output ports are called output queued switches. These may take the form of a single shared memory comprising virtual buffers as described above or as a separate memory for each output port. In either case, an N input switch has a possibility of N packets being destined for a given output at each time step. Therefore, in order to avoid packet loss, output queued switches are designed to buffer N packets at a time for each output port and deliver one packet at a time to each output port. This architecture achieves optimal delay and throughput performance and head-of-line blocking immunity at the cost of large and fast buffering requirements [27]. One

notable example of an output queued switch is the AT&T Knockout Switch [28] which is a fully connected switch that uses packet filters, concentrators, and shifters to load balance data delivery to the output ports.

2.3.4 Input Queuing

Input queued switches use a single buffer at each switch input. In contrast to the output queued switch, the buffers in an input queued switch only need to accept packets from one source each and provide packets one at a time to the switch. This is very advantageous because the memory bandwidth requirement is reduced to twice the line rate, R.

$$BW_{mem\text{-}IQ} = 2*R \qquad \text{Eq. 6}$$

Traditionally, there has been a significant and well-known disadvantage to input queuing, namely head-of-line (HOL) blocking. HOL blocking occurs when there is a packet in a first-in-first-out (FIFO) input queue that cannot be transmitted even though the route through the fabric to its destination is clear because a packet at the head of the input queue cannot be transmitted due to output (or fabric) contention. From a study of queuing theory, HOL blocking has been shown to reduce the throughput of an input queued switch to approximately 59% of the offered load [29]. Fortunately, there is way to take advantage of the reduced memory bandwidth required by input queuing while avoiding this throughput penalty. The use of *virtual output queues* as introduced in [30], and an appropriate scheduling algorithm such as iSLIP described in [31] can increase the throughput to 100% for uniform traffic. In this method, FIFO input queues at each input port are replaced with memories containing a virtual queue for each output port of the switch. These queues are serviced according to a round-robin algorithm and since each virtual output queue

contains only packets destined for a single output port, there can be no HOL blocking. Input queuing is commonly used in current commercial switches using virtual output queues and such a scheduling algorithm.

2.3.5 Combined Input/Output Queuing

A typical implementation of a combined input/output queued switch consists of FIFO input buffers added to an output queued switch or a shared memory switch. This allows input contention and output port contention to be handled by separate arbiters. Variations including replacing the input FIFO with a VOQ and handshake scheduling have also been proposed to improve throughput performance [32]

The benefit of using a combination of input queues and output queues in the switch architecture is in achieving the throughput performance of an output queued switch without the memory bandwidth requirement dependence on the number of ports. This, in turn, allows scaling of CIOQ switch architectures to greater numbers of ports than is practical in and output queued switch. Additionally, HOL blocking, which can limit the throughput of a strictly input queued switch, is avoided. It has been shown that with a speedup of two, that a CIOQ switch can behave identically to an output queued switch at the expense of a more complicated scheduling algorithm [33]. Continued research into this architecture is currently underway and it is believed to be one of the most promising architectures in terms of creating larger switches. For instance, in [34], a hybrid switch architecture is presented where the use of input queuing is combined with a buffered crossbar and support for variable sized packets to yield a switch with 300 Gbps throughput performance.

2.4 Routing Methods

Routing refers to the selection of the path through the network from the source to the destination. The routing algorithm has implications on the packet delay, packet transit time, buffer management, and required buffer sizes. It can be classified according to how routing decisions are made, what factors are taken into account in choosing the route, and how it is implemented.

Routing decision types include source routing, distributed routing, and centralized routing. A combination of these types may also be implemented. In source routing, the selection of the route through the network is made a priori by the sender and included at the start of the message. At each intermediate routing node, this information is examined in order to control the routing decision. Examples of source routing implementations include the high-performance local area network, Myrinet and the research wide-area optical network, Blazenet [35][36]. In contrast to source routing, distributed routing disperses the routing decisions among the intermediate nodes. In this case, the source provides minimal information (i.e. the destination address) and the nodes along the route determine the next link to propagate the message on the fly based on their own knowledge of the network. Global information is typically not required by the intermediate nodes in distributed routing. One advantage of distributed routing over source routing is that the route information does not have to be transmitted along with the message, thereby conserving bandwidth. In centralized routing, a central control unit, which is independent of the source and intermediate nodes, is responsible for making the routing decisions. Centralized routing is common in single-instruction multiple-data (SIMD) machines [37].

Routing algorithms may be deterministic or adaptive depending upon what is taken into account when making routing decisions. A routing algorithm is

deterministic if the path is uniquely determined by the source and destination, whereas adaptive routing takes into account variable conditions such as congestion [24][23]. Since adaptive routing schemes need to gather information about the state of the network, there is a communication overhead and implementation complexity associated with them. Additionally, if the state of the network is changing rapidly, there is a potential for routing decisions to be based on out of date information, and as such, adaptive source routing is most applicable in networks where traffic conditions change slowly. Routing algorithms may allow a packet to go backwards (in the case of downstream congestion) in order to find a new route through the network and they may also allow data to be routed in a direction that is not strictly toward the destination (for instance in a mesh network). These characteristics are called backtracking and non-minimal respectively and are only applicable to adaptive routing algorithms because deterministic algorithms will always select the same path, which will, hopefully, be the shortest path possible [37].

Whether by source or distributed routing and deterministic or adaptive routing, the routing algorithm needs to be implemented in hardware or software. Two common approaches are the use of lookup tables and finite state machines to either lookup the complete path or the next outgoing link or to compute the path.

2.5 Scheduling

Scheduling is the control process by which data is taken out of the switch queues at points of contention. This includes both the allocation of the output port bandwidth to packets from multiple sources and multiple priorities. A feature of modern commercial switches that is gaining importance is the support of quality of service (QoS) features. Two important QoS parameters are bandwidth and latency

guarantees. Others include availability, jitter (in terms of packet delivery), loss characteristics, and traffic priority. QoS features are implemented within switch architectures by including multiple queues (e.g. for different priority levels) and by using a scheduling algorithm that services queues such that these parameters can be guaranteed.

Scheduling is most often based on some type of round-robin servicing of queues. Implementations include standard round-robin, priority round-robin, weighted round-robin, and iterative approaches. With standard round-robin scheduling, each queue is serviced in order one after another. Priority round-robin adds priority to some queues and weighted round-robin services queues in order, but gives multiple "turns" to some queues in the rotation. A scheduling algorithm introduced to increase the realizable bandwidth of input queued switches from the well-known (empirical) $2-\sqrt{2}$, or $\approx 58.6\%$, to near 100% is called iSLIP. This method uses virtual output queues to mitigate head-of-line blocking and is based on iterative round robin packet serving with a modification to when the grant pointer is updated. This algorithm has been implemented in both commercial and research switches [31]. Another scheduling algorithm, FIRM, was designed for distributed scheduling [38]. The industry trend however is toward custom and proprietary scheduling algorithms [26].

2.6 Fabric Organization and Scaling

In order to increase the size of a switch, levels of hierarchy are introduced where sub-blocks are connected to form larger blocks, which are in turn connected together to form a switch "fabric". At the lowest level, most switch fabrics are made up of the same primitive circuit: a "crossbar" [34]. These cells are then connected

together in a variety of ways to form a switch. Large switching fabrics can be formed that connect many nodes in what are known as multi-stage interconnection networks (MIN). There are, of course, trade-offs in the implementation at each level of hierarchy.

2.6.1 Switching Elements

A common base element for building electronic switches is the crossbar. An electronic crossbar is minimally a two-input and two-output switch element that can connect each input to either or both outputs as illustrated in. This can be scaled to an NxM switch by using more switch elements. The name comes from the old telephone switches, which were electromechanical devices capable of making straight-through connections or crossed connections. There are two common implementations of electronic crossbars: crosspoint switches and multiplexor-based switches. Although the term “crossbar” and “crosspoint” are often used interchangeable, the distinction between crosspoints and multiplexors will be maintained here as two types of crossbar and described in the following paragraphs.

A switch formed by small switch elements placed at the intersection of interconnect lines will be referred to as a crosspoint switch. An NxM crosspoint switch can be made by routing metal lines in an NxM grid with N horizontal lines for inputs and M vertical lines for outputs (or vice versa). At the intersection of each of these lines a two-position switch (open or closed) allows any input to be connected to any output. Fan-out is also possible with this configuration, which allows for multicast and broadcast, but the undesirable possibility of connecting multiple inputs to the same output must be guarded against by closing at most one switch on a given output line.

A multiplexor-based crossbar can be implemented electronically by using two two-to-one multiplexors. This may be realized with logic gates, transmission gates, or tri-state buffers. Larger crossbars can be made by using multiplexors made from higher fan-in logic gates or by using multiple stages of smaller multiplexors in a tree structure. There are performance trade-offs in both cases. Using higher fan-in logic gates can lower the transistor count for larger multiplexors, but it also increases the delay. As such, the availability of high fan-in gates within a standard cell library generally limits the single stage multiplexor size to around four inputs. In order to make larger multiplexors for a crossbar switch, a multi-stage tree structure is used. Using this approach increases the logic depth and, therefore, also increases the delay through the switch. For any appreciable sized multiplexor, such a tree structure will be required and one tree with N inputs will be needed for each of the M outputs in an NxM crossbar.

As a result of the structure of a crosspoint switch, a total of M*N control lines are required to configure all of the switch elements. A decoder can be used to set these lines from a binary-encoded control word and at the same time prevent multiple inputs from driving the same output. A multiplexor-based crossbar on the other hand uses a binary encoded control word and inherently does not allow multiple inputs to connect to one output. An additional advantage of a multplexor-based crossbar is that it lends itself well to synthesis from a standard digital library.

2.6.2 Scaling Switch Fabric

Increasing the number of ports for a switch fabric generally requires increasing the number of smaller switch elements to form a multistage network. Depending upon how this network is constructed it can be blocking or non-blocking.

There are a number of classic architectures that have been used to create multistage networks. Among these are banyan networks and benes networks [37]. The most famous multistage interconnection network that is non-blocking was proposed by Charles Clos in 1953, which has been studied and utilized extensively [39][40]. The Clos architecture consists of an odd number of stages of smaller switches. To form a three-stage Clos network with N=n*r ports, r switch elements in the first and third stages which are n x m and m x n respectively are connected to m second stage switches which are each r x r. For such a network to be strictly non-blocking, the condition: $m > 2(n - 1)$ must hold. This feature can be used to build arbitrarily large non-blocking switches out of smaller ones.

Chapter 3

VIVACE SWITCH ARCHITECTURE

The switch design presented here is a forward-looking implementation of a multi-port, multi-gigabit packet switch. The novelty of the design lies in its ability to overcome bottlenecks common to current commercial data switching systems and demonstrate a path to even greater scalability. A fundamental influence on the design of this switch and the means by which it can overcome many bottlenecks is the high number of data inputs and outputs included based on the use of optical inter-chip signaling.

This chapter will provide an introduction to the overall switch design architecture and goals, and compare its features to those found in various switch classifications described in Chapter 2. The detailed architecture and functionality will be given including the network protocol, dataflow, and logic implementation.

3.1 VIVACE Switch Architecture Introduction and Performance

The overall system architecture will be considered here briefly as a contextual background for the description of the Switch ASIC to follow. The primary components of the full VIVACE system concept consisted of multiple standalone computer workstations each with a fiber-optic communication link to and from a central switch module. Each of the workstations would operate as a part of a distributed parallel computer to accelerate a computationally complex calculation. Intense inter-processor communication would be sped up by the high-bandwidth

available through the central switch module. The switch module itself would consist of a multi-chip module (MCM) mounted on a printed circuit board with other appropriate support electronic hardware. This includes optical-to-electrical converters, serializer/deserializer (SERDES) devices, and FPGAs that perform interface functionality as shown in Figure 1. Its intended operating environment within a cluster of high-performance computer workstations influences the switch design for the VIVACE program.

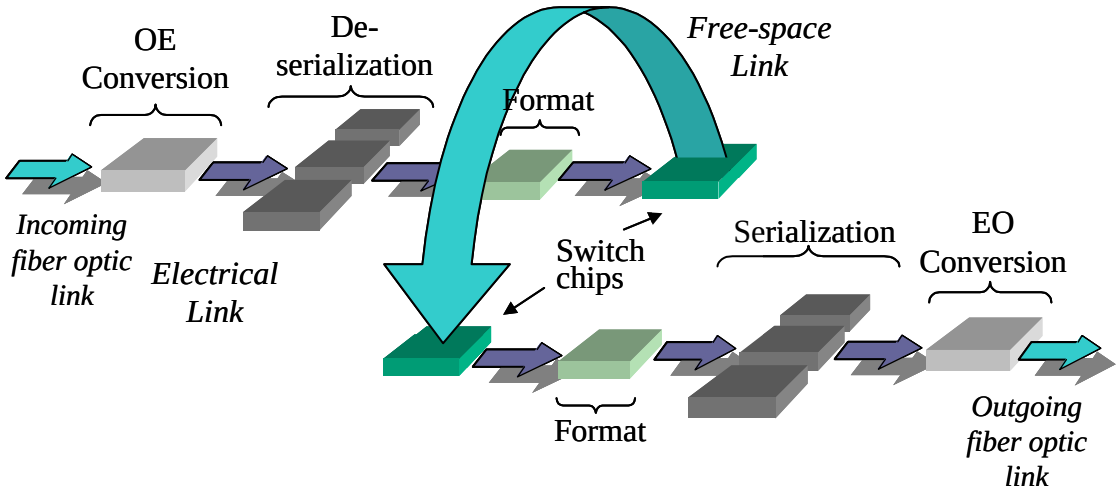

Figure 1. Dataflow through switch (one path, one direction shown).

Each workstation, or Host, has an optical network interface card (NIC) that allows it to communicate with the central switch. This NIC is a custom add-on card that uses the peripheral component interface (PCI) to transfer data to and from the application memory. It was developed specifically for the VIVACE program and because it uses an optical interface it is referred to herein as the VONIC. In order to

provide a high-bandwidth, low-loss, and low connection-count link to the switch, a twelve-channel optical fiber interface is used to transfer data from the NIC to the switch. Custom logic is used to transfer data received from the PCI bus to SERDES inputs and vice-versa.

Data arriving at the central switch is received on a motherboard containing interface electronics and the optically interconnected switch MCM itself. The serial optical streams are converted back to parallel electrical signals at the edge of the motherboard and then FPGAs are used to format these parallel signals into the port format of the switch. As such, the switch motherboard functions similarly to line cards in a traditional data communications switch, however, the physical implementation is much different.

The MCM is mounted on the motherboard and consists of multiple, fully interconnected Switch ASICs. Each port of the switch is assigned to a specific ASIC on the MCM. These individual ASICs handle the switching of data between ports according to a custom protocol developed for this purpose.

A number of goals were set for the custom protocols and switch design that would set it apart from other switch implementations. One of the primary strengths in the VIVACE design was the rich interconnect available within the switch module as a result of the use of free-space optical links between the multiple Switch ASICs. The fully interconnected nature of the optical system leads to the ability to perform efficient multicasts and broadcasts. True concurrency between multiple messages being transmitted through the switch, by eliminating shared control paths, was desired along with low message overhead. Low latency and the ability to handle arbitrarily long messages was sought to enhance the end application performance.

Finally, hardware support for flow control including error and overflow negative-acknowledge generation was desired to reduce the burden on compute nodes connected by the switch. Through the development of the custom switch-level protocol and logic implementation, these goals were met.

3.2 Detailed Architecture and Functionality

The focus of this dissertation with regard to the switch design and implementation is on the ASIC itself with supporting details from other parts of the overall switch and network given as needed. This Switch ASIC corresponds to what is referred to as fabric device in commercial switches. Several Switch ASICs are combined to form a larger switch fabric. These ASICs are placed on a single MCM to form what is analogous to a fabric card in commercial switches.

For modularity and scalability, the switch functionality is broken down into switch cores which can be replicated within a given Switch ASIC. The primary logic function of the Switch ASIC core can be concisely stated as accepting a data stream at an electrical input port and sending it to a subset of optical output ports while, at the same time, accepting a number of data streams at optical input ports and sending one of them to an electrical output port. The combination of multiple Switch ASICs performing this function allows for a multi-port, free-space optically interconnected switch to be constructed. As the VIVACE program progressed, a number of factors were encountered that forced reducing the number of switch cores that could be included on a single Switch ASIC as well as the number of Switch ASICs that could be combined onto a single MCM. The end system design was for each ASIC to contain one switch core, and thus implement one switch port, and

combine eight Switch ASICs onto one MCM to form an eight port switch. The sections that follow describe the design and operation of a single switch core.

3.2.1 Overall Architecture

The logic required to handle data routing and control packets in the Switch ASIC can be logically divided into two primary blocks. These are the Inward Logic and Outward Logic blocks. The Inward Logic consists of the functional blocks that interface with the electrical input port and control the optical output ports. Conversely, the Outward Logic contains functional blocks to receive data from the optical inputs and handle sending data out the electrical output port as illustrated in Figure 2. There is communication between the Inward Logic and Outward Logic on a given Switch ASIC. The electrical interface to the switch has been described above. The optical interface consists of a multi-bit optical path from a given Switch ASIC to all Switch ASICs on a single MCM. This communication is illustrated in Figure 3. The hardware description language VHDL has been used to implement all of the logic for the Switch ASIC. As such, many parameters, including the width of optical paths, and especially, the number of ports in the full switch have been abstracted such that they can be easily changed. The width of the optical path matches the electrical input and output port and consists of 32 links for each optical data port. A 33rd optical link is included with each optical port that serves as a control marker. Within the Inward and Outward Logic of the switch core there are data flow paths, control logic, and input/output interface blocks. There are additionally protocol-specific signals which allow communication between the Inward and Outward Logic blocks of a given switch core.

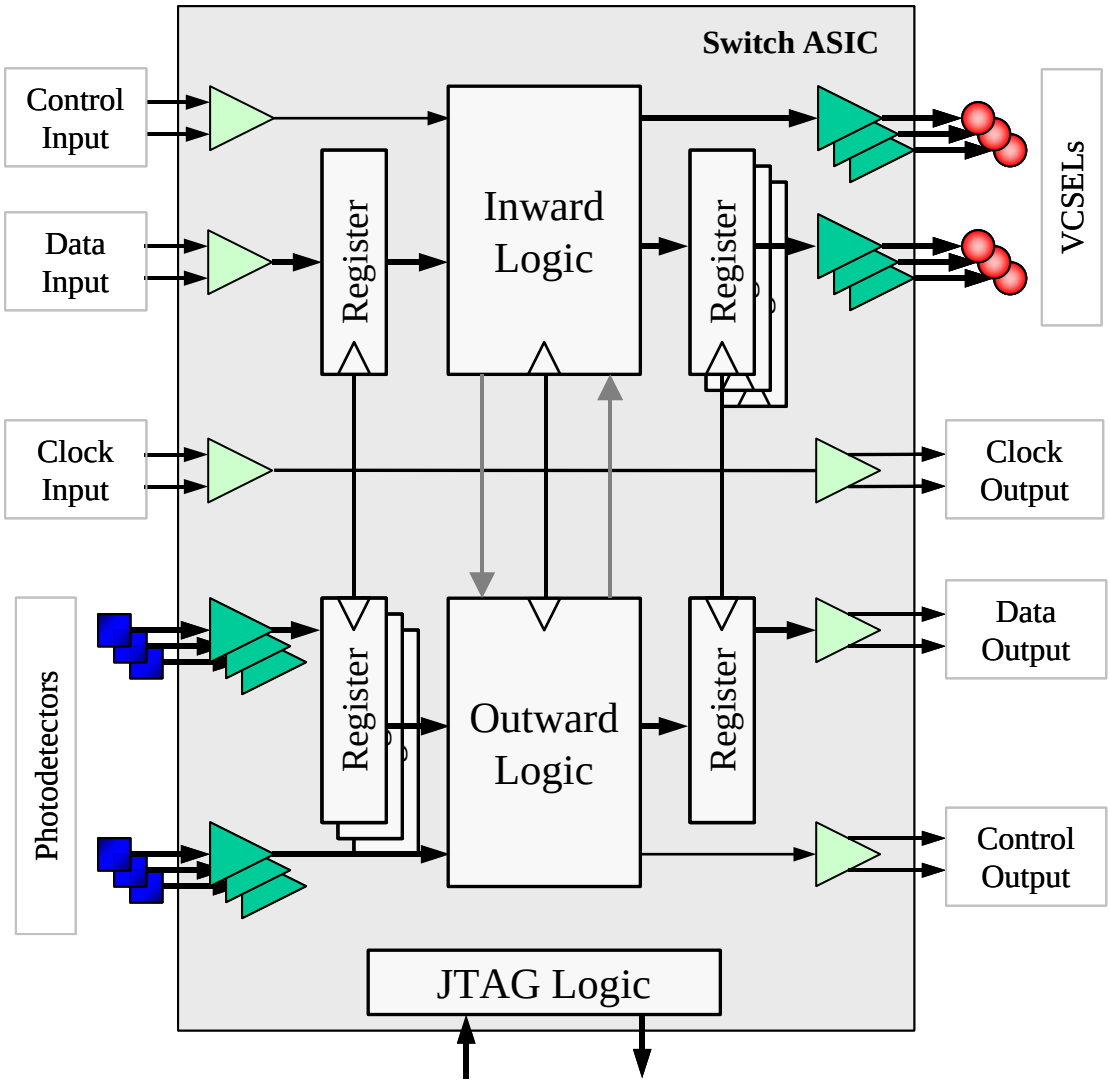

Figure 2. Switch ASIC top-level block diagram.

The terms *inward* and *outward* were used throughout the VIVACE network to describe operations from host to host. Since the switch is at the center of the system and the mirror effecting the folded optical interconnect is at the center of the switch module, data moving toward the mirror has be described as traveling

inward, and conversely, data moving away from the mirror has been described as traveling outward. These terms are similar to the terms *ingress* and *egress* often used to describe switch ports except that they have been applied to all parts of the end-to-end communication from one host to another. For simplicity, in the discussion that follows the Inward Logic and Outward Logic will often be referred to as "IL" and "OL", respectively and will refer to logic within the switch core unless otherwise stated. Additionally, the receive side of a logic block is in some cases abbreviated as "Rx" and the transmit side as "Tx".

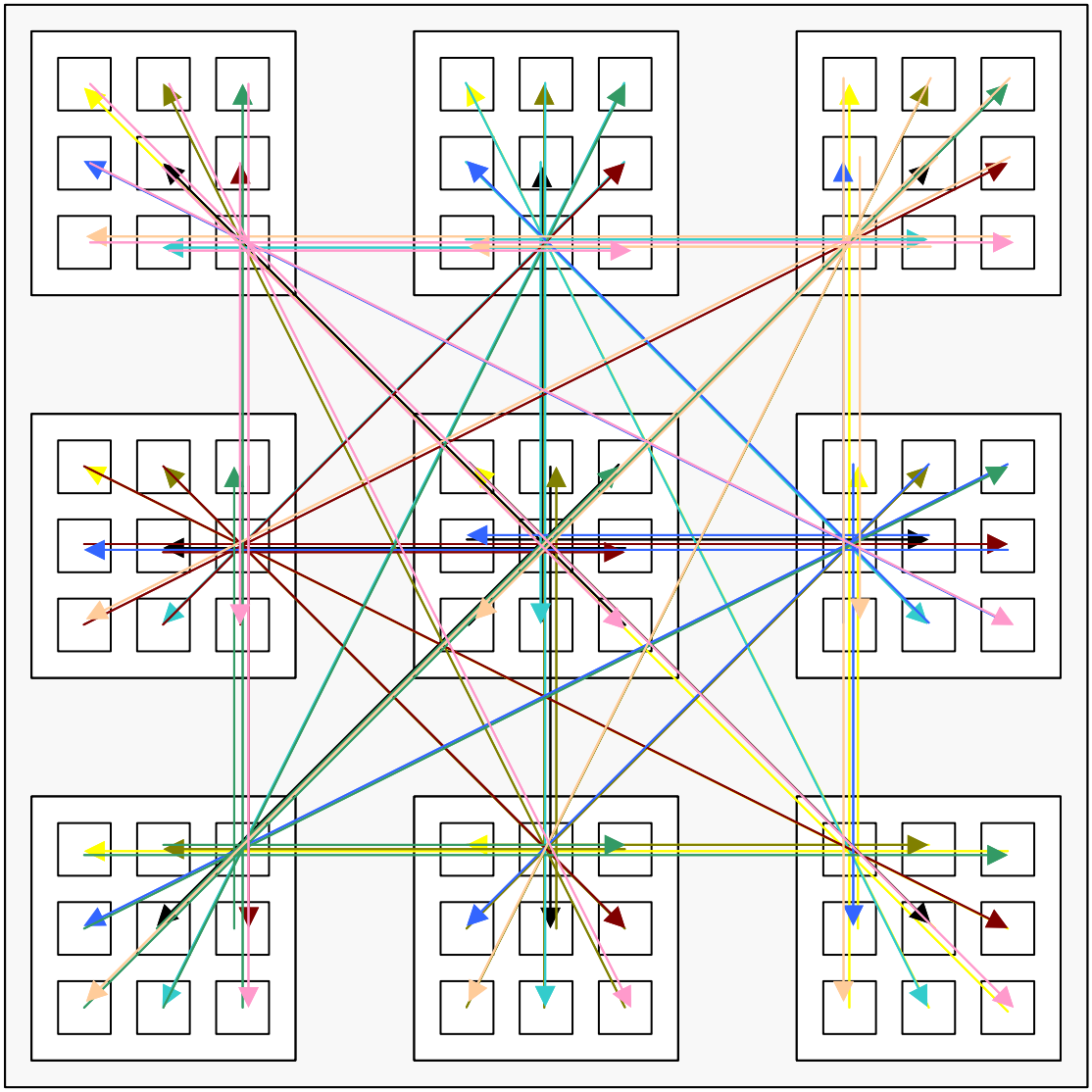

Figure 3. Optical connectivity diagram for nine chips. Each chip has nine ports and is placed on a single MCM. Note that each arrow represents a wide optical path.

3.2.2 VIVACE Network Protocol

From the standpoint of the protocol, the interface to the switch core consists of a 32-bit data word, a single bit control marker, and a clock. This interface

is replicated for both the electrical input and electrical output. All data and control information is presented to the switch port on the 32-bit input bus while the control marker indicates to the switch whether the word at the input is a data word or one containing control information. Routing of data through the switch is done according to the data-packet header sent at the start of message transmission. The switch output is placed on the 32-bit output bus with a marker to indicate control versus data. This interface is designed to be source-synchronous to simplify clock synchronization and thus the clock input and output are used to load and unload the data bus and control marker. The width of the input and output busses comes from the desire to have a port data rate of approximately 10 Gbps and the desire to process data within the switch at an aggressive rate of 300 Mbps without the added complexity of serialization and de-serialization within the Switch ASIC, which is instead done on the switch Motherboard. Figure 4 shows the division of the standard and custom protocols used in the VIVACE network.

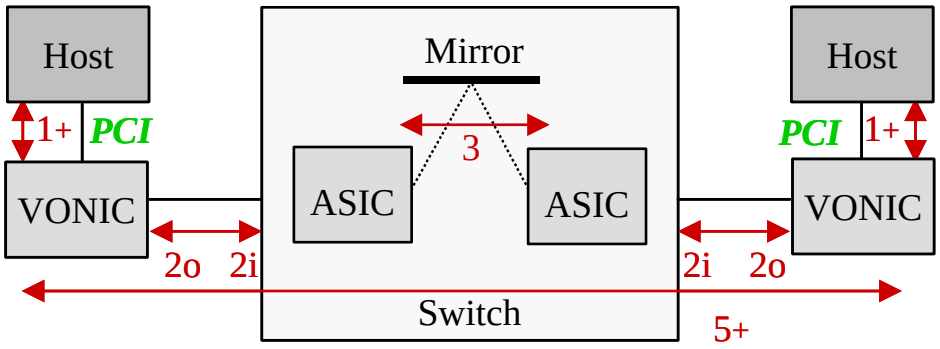

Figure 4. Protocol naming convention by network location.

The data into and out of the switch module is in the form of twelve serial streams for each port. AMCC serial backplane devices at both ends of the link accomplish the multiplexing of data onto these serial lines [41]. These devices rely on 8-bit to 10-bit (8b/10b) encoding to maintain synchronization without transmitting a clock along with the data. As a result of the 8b/10b encoding there are twelve out-of-band characters that cannot occur in a valid data stream and can thus be used to communicate control information. These "K-characters" are listed in Table 1 and are used by the VIVACE protocol to pass control information through and to the switch. Since data flows through the switch in a synchronous manner as parallel words, the 8b/10b encoding and associated overhead that is needed for the serial links to and from the switch is not required within the switch fabric. Instead, the control marker bit indicates whether a word is to be handled as data or as control. The switch word size of 32 bits makes it possible for the control words to carry a payload.

Table 1. Pre-defined Out-of-Band Characters

K-Character	**8-bit representation (7:0)**	**Protocol Usage**
K28.0	0 0 1 1 1 0 0 0	nakBusy
K28.1	0 0 1 1 1 1 0 0	Pad
K28.2	0 0 1 1 1 0 1 0	nakErr
K28.3	0 0 1 1 1 1 1 0	SOM (protocol 3)
K28.4	0 0 1 1 1 0 0 1	rsvd
K28.5	0 0 1 1 1 1 0 1	Idle (protocol 2)
K28.6	0 0 1 1 1 0 1 1	rsvd
K28.7	0 0 1 1 1 1 1 1	Management
K23.7	1 1 1 0 1 1 1 1	rsvd
K27.5	1 1 0 1 1 1 1 1	rsvd
K29.7	1 0 1 1 1 1 1 1	rsvd
K30.7	0 1 1 1 1 1 1 1	rsvd

3.2.2.1 Features

The first and foremost feature of the network protocol described here is, of course, its ability to forward packets to the proper destination. However, the simplified design allows for low message overhead and latency, which is beneficial in distributed applications. Additionally, greater data throughput can be achieved by the ability of the protocol to handle large packets, resolve control packet contention, perform hardware-based multicast, and carry out back-to-back transmission with little or no message separation.

3.2.2.2 Data Flow

In the simplest case, a message is sent through the switch by sending a single control word that contains the destination address for the packet followed by an arbitrarily long block of data. In addition to the address, a number of other control words are defined to handle a variety of more complicated situations. The primary control words are listed in Table 2. Other control words for interfacing with the switch module itself, for purposes such as setting up configuration registers or reading status registers, were also defined as well as control words associated with connecting multiple switches together via a host acting as a bridge. According to the VIVACE protocol, it is not necessary for a host that wishes to send a message to first ask or be granted access to the network. Built-in collision detection and flow control feedback in the switch hardware alleviates this need in order to increase the best-case throughput without negatively impacting performance in other situations.

Table 2. Primary Data and Control Words

Name	Words	Control	Description
Idle	1	1	Character sent when no other control or data
Message	1 to A	0	Address bit vector(s); A ≥ 1
	A+1 to N	0	Data word(s); N ≥ A+1
Nak	1	1	Negative Acknowledge
Pad	1	1	Null character inserted to delay data stream
Mgmt	1	1	Flow control and switch management

3.2.2.3 Control Set

The Idle character is a single control word used when there is no message or other control word to be sent. In protocols 2i and 2o this is the fiber channel K28.5 character repeated for all four bytes of the word. This places all of the serial channels making up a switch port into an idle state. Within the switch, the Idle character consists of all zeros such that the VCSELs associated with idle links are not emitting beyond their bias condition.

A data message is composed of one or more address words followed by one or more data words. The ability to handle packets with multiple address words at the start of a packet allows for multi-stage switches to be constructed from the VIVACE switch. The switch uses the first address to route the packet, stripping it out of the packet such that wormhole-type routing could be done. The address itself is a bit mask. This simplifies the switch design, as the address does not need to be decoded in order to route the packet. Scalability is not impacted in the current design because only eight bits are necessary and thirty-two bits are available within a single address word, meaning that a switch of up to thirty-two ports could be constructed in this manner without the need to encode the address or use multiple address words.

Data words following the address can be of arbitrary length in order to allow low message overhead for large messages.

The purpose of the Nak control character is to provide flow-control feedback within the switch and between the switch and host. Two types of negative acknowledge characters have been defined. One, a nakBusy, is used when a point in the path to the destination port of a packet is already in use, thus blocking transmission. A K28.0 character in the most significant byte of the word indicates the nakBusy. It does not carry a payload and so the remaining three bytes are unused and set to zero within protocol 3. The other Nak character, a nakError, is used to notify the sender that a packet was not properly received due to a transmission error. A nakError would most often be sent after a packet has been completely transmitted as a result of a cyclic redundancy check (CRC) failure at the destination. Therefore, the routing information for a nakError is not implied as in the case of the nakBusy. To provide routing information for nakError words, the payload area of the control word is used to hold the packet origination address.

There are several management characters used in the VIVACE protocol. They are formed by a K28.7 character followed by a payload that indicates the type of management character and optional data. One of the management characters is Throttle. Pad characters and Throttle characters are coupled flow control words. In the event that a point in the path to the destination, either within the switch or at the network interface, cannot handle the incoming data rate a throttle request can be sent back to the sender. This throttle request indicates the length of pause required by providing a number of Pad characters to be inserted. The Pad character is used as a special control character that does not terminate the packet being transmitted, but

delays its transmission. Pad characters are discarded at the point where they were requested.

Two other management characters are used for port initialization. These are the Who-am-I and UR characters. In order for a host or network interface card to generate return addresses, it must know which port on the switch it is connected. This means that the switch itself must assign or be aware of port numbers. The Switch ASICs making up the switch module must be identical and so hard coding port numbers into the silicon die is not practical, but this can still be handled in a number of ways. Chip identification could be hard coded onto the MCM by wirebonding each chip to a different fixed value but this takes extra bond pads which is not efficient. Port numbers could also be assigned and written into configuration registers within the Switch ASIC at power-on, but again this adds complexity that is not desirable. As an alternative approach, the switch protocol takes advantage of the fact that due to the optical interconnect pattern, each Switch ASIC inherently knows which port each other Switch ASIC is connected to and can relay this information. Hence, the Who-Am-I and UR ("you-are") control packets are used to request port information and reply to the requester respectively. Who-Am-I requests are sent as broadcast messages, which offers great redundancy against traffic collisions due to an erroneous port number assignment. The UR reply is sent to the requesting Switch ASIC with the appropriate port number in the payload section of the control word based on the optical channel where the Who-Am-I request was heard.

Throttles are an important part of the switch functionality and will be considered further here. There are provisions within the protocol for throttle requests to originate from two locations: the Outward Logic first-in first-out buffer (FIFO) and

the VONIC outward FIFO. Though not discussed in this section, the VONIC logic is also organized into inward and outward logic corresponding to data traveling to the switch or back from the switch. FIFOs within the VONIC allow it to interface between the switch and the PCI bus. The precaution of hardware generated Throttle requests is built in because the Outward Logic and VONIC FIFOs have the potential for data loss. Depending upon where they originate, Throttle characters are handled slightly differently. Table 3 and the series of drawings in Figure 5 depict how Outward Logic generated throttle requests are handled. Two VONIC cards and two Switch ASICs (abbreviated SWIC) are depicted in this figure. Blue and green boxes represent Inward and Outward Logic, respectively. Black arrows represent data flow and green arrows represent control information.

Table 3. Steps in Resolving Outward Logic FIFO Filling

Step	Action
1	Host-A is transmitting data to Host-B
2	SWIC-B Outward FIFO becomes almost full. (This FIFO is receiving data from the SWIC-A Inward Logic and sending data to the VONIC-B.)
3	SWIC-B Outward Logic tells the SWIC-B Inward Logic to send a Throttle packet by sending an InsertThrottle signal
4	SWIC-B Inward Logic interrupts its inward data stream (if any) and sends a Throttle packet to SWIC-A Outward Logic. Interruption is only necessary if the throttle needs to be sent to the same destination as the current data stream.
5	SWIC-A Outward Logic interrupts its outward communication to VONIC-A (if any) and forwards the Throttle to VONIC-A Outward Logic.
6	VONIC-A Outward Logic sends an InsertPads to VONIC-A Inward Logic
7	VONIC-A Inward Logic sends the prescribed number of Pads back to SWIC-A Inward Logic (thereby throttling its message to Host-B as desired).
8	SWIC-A Inward Logic forwards this Pad to SWIC-B Outward Logic
9	SWIC-B Outward Logic discards these Pads from SWIC-A Inward Logic (This Pad is not needed by VONIC-B – the data transfer from SWIC-B Outward Logic to VONIC-B Outward logic continues relieving the FIFO in SWIC-B Outward Logic.)

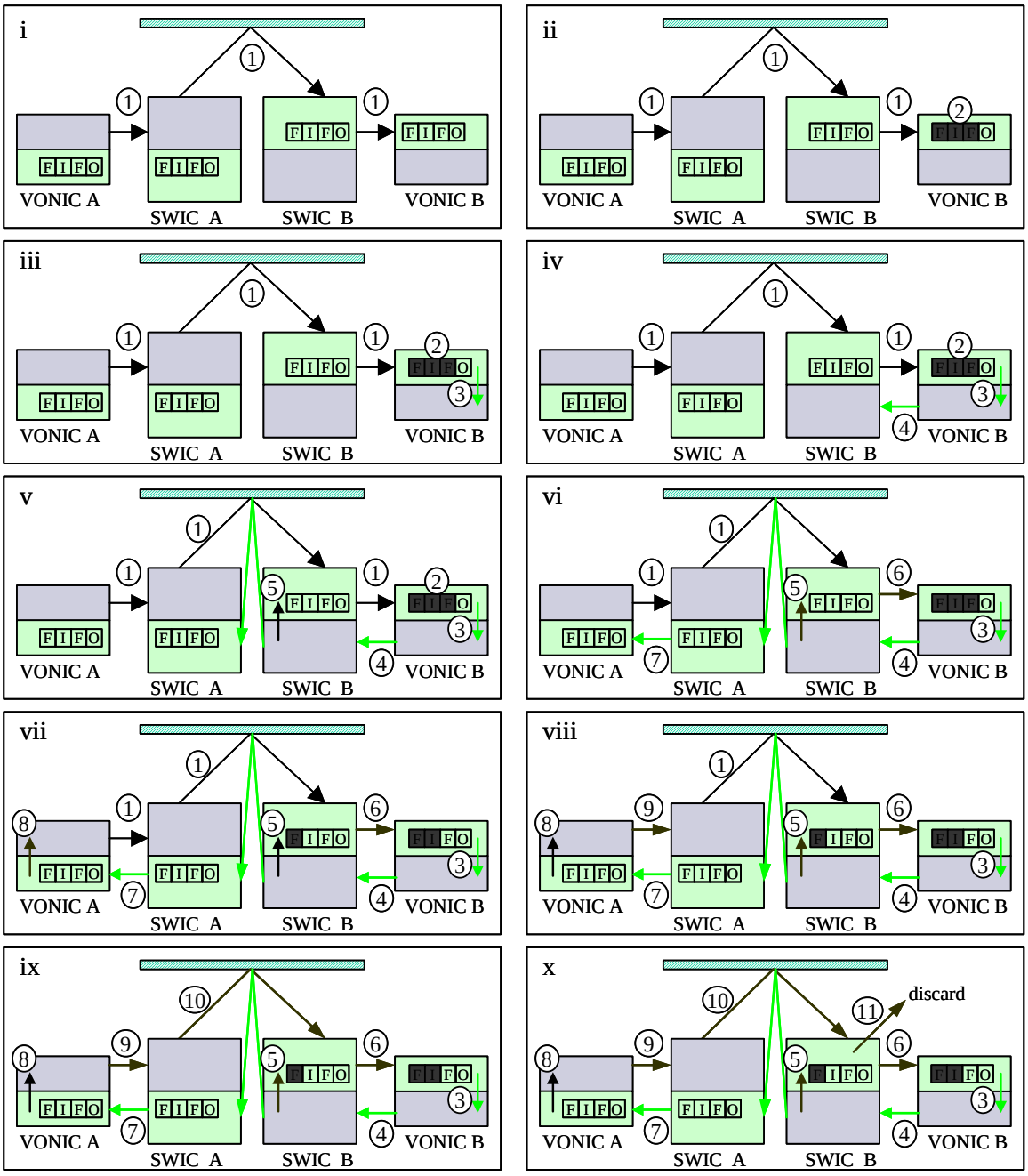

Figure 5. Handling of VONIC-generated Throttles.

Similar steps are taken to resolve the problem of the VONIC's FIFO getting full. In this case, a control packet must be sent from the VONIC to the switch where it is detected by the Inward Logic and immediately causes the Outward Logic to begin sending Pad characters to relieve the VONIC FIFO. In this case, the Throttle character must also be routed to the sending VONIC so that the Outward Logic FIFO does not overflow as a result of pausing the switch-to-VONIC data transmission. This process is depicted as before in Table 4 and Figure 6.

Table 4. Steps in Resolving VONIC FIFO Filling

Step	Action
1	Host-A is transmitting data to Host-B
2	VONIC-B Outward FIFO becomes almost full. (This FIFO is receiving data from the switch and sending data to the Host.)
3	VONIC-B Outward Logic tells the VONIC-B Inward Logic via an InsertThrottle to send a Throttle packet
4	VONIC-B Inward Logic interrupts its inward data stream (if any) and sends a Throttle packet to SWIC-B Inward Logic.
5	SWIC-B Inward Logic 1) decodes Throttle from VONIC-B and sends InsertPads to SWIC-B Outward Logic and 2) forwards the Throttle to SWIC-A Outward Logic via optical data link (interrupting any data communication taking place if present)
6	SWIC-B Outward Logic begins sending Pad characters to VONIC-B Outward Logic and continues until it has sent the prescribed number of them. The data that is being received at the SWIC-B Outward Logic is stored in the FIFO.
7	SWIC-A Outward Logic forwards this Throttle to VONIC-A Outward Logic
8	VONIC-A Outward Logic sends an InsertPads to VONIC-A Inward Logic
9	VONIC-A Inward Logic interrupts its data transmission toward B and, until it has sent the prescribed number, sends Pad characters to SWIC-A Inward Logic.
10	SWIC-A Inward Logic forwards this Pad characters to SWIC-B Outward Logic
11	SWIC-B Outward Logic discards these Pad characters from SWIC-A Inward Logic (Pads are already being sent from SWIC-B Outward Logic to VONIC-B Outward Logic from (6) above.)

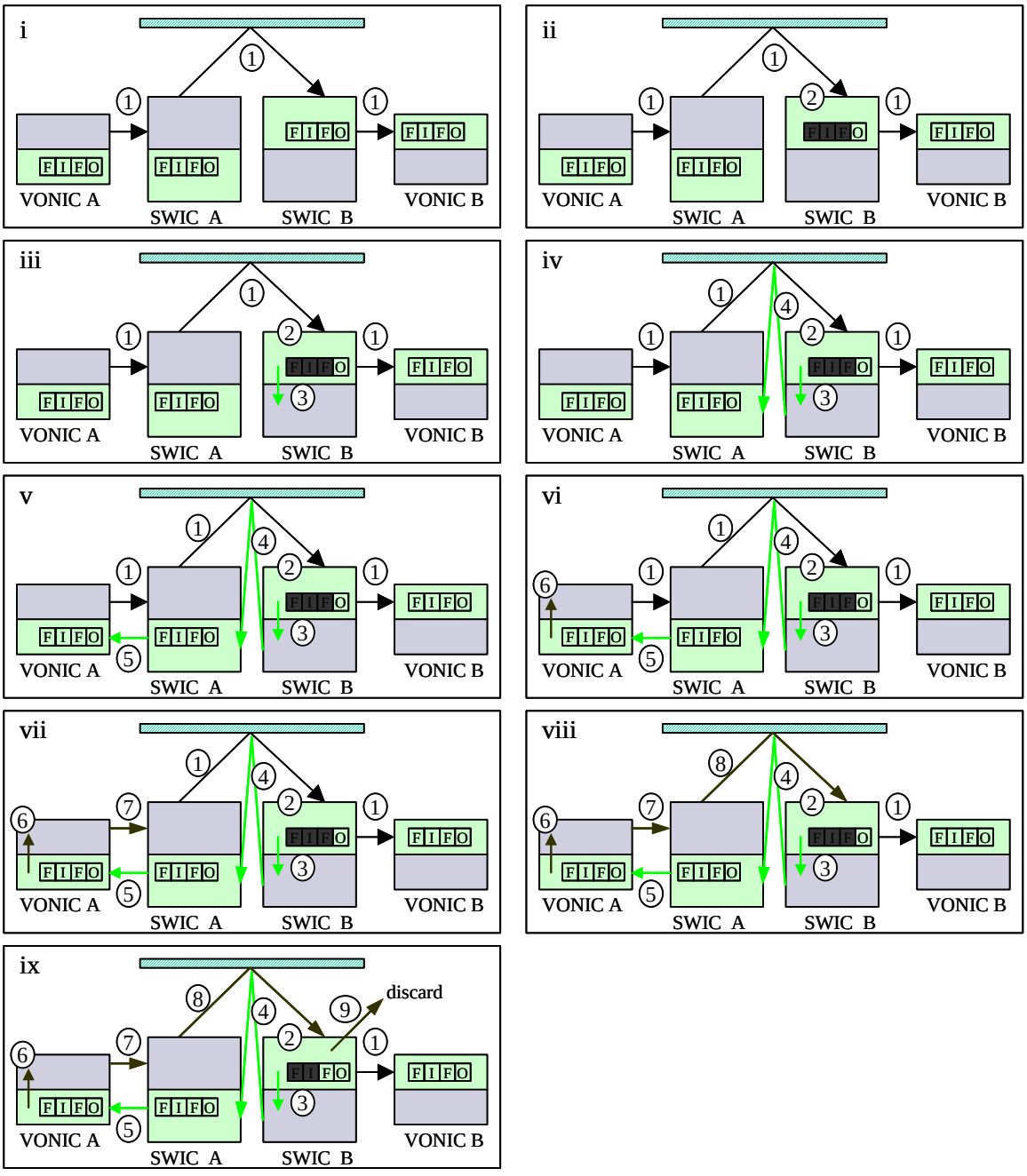

Figure 6. Handling of Switch OL-generated Throttles.

3.2.3 Inward Logic

The responsibility of the Inward Logic is to accept data from the electrical input port, recognize control packets that must be handled, resolve contention for the optical output ports, and send data and control words to the appropriate optical output ports. It is further subdivided into receive logic and transmit logic. The receive logic is associated with the electrical input port and there is one copy of it per chip. The transmit logic within the Inward Logic block is instanced once per optical output port and is responsible for taking data from the Inward Logic data first-in-first-out (FIFO) buffers and sending it to the VCSEL drivers.

3.2.3.1 Inward Logic Block Diagram

A block diagram of the data path through the Inward Logic is shown in Figure 7. Boundary scan is used both in the Inward Logic and Outward Logic for testability. This boundary scan chain includes both the electrical inputs and outputs of the Switch ASIC as well as the optical I/O. Additionally, all electrical and optical I/O are registered to reduce timing ambiguity. For normal data or control packet transmission, words sampled from the electrical input bus are used by the receive side of the Inward Logic.

The receive block of the Inward Logic tracks the state of the link from VONIC to switch and decodes control information from the incoming data stream. It also provides routing information for the decoded control words to the transmit side of the Inward Logic so that access to the optical links can be allocated.

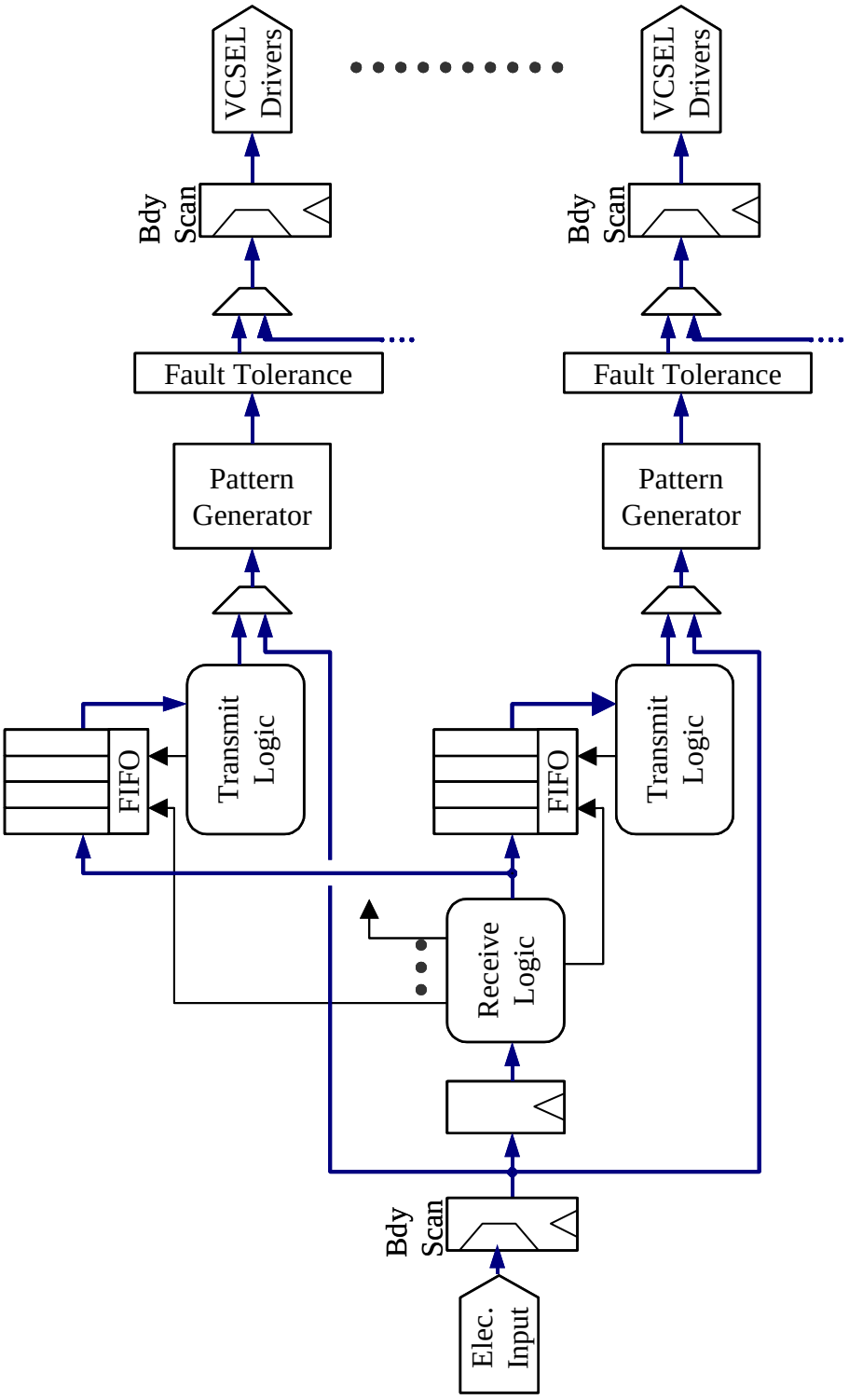

Figure 7. Inward Logic block diagram.

The following control words are decoded and used by this block. Idle characters are decoded as they indicate whether the VONIC-to-switch link is active or idle. The Idle character itself is not actually passed on, but rather will be generated if appropriate. This logic block must decode Throttle characters generated by the VONIC. In order to effect the throttling process, upon decoding an incoming Throttle packet a signal is sent to the Outward Logic and the Throttle is transmitted optically, bypassing the Inward Logic FIFO. The Who-Am-I control packet is also decoded by this receive logic. Broadcast of the Who-Am-I request is handled differently than a data broadcast because it is a control word. NakError packets are decoded at the input so that an appropriate protocol 3 Nak message can be sent.

The receive block of the Inward Logic uses a state machine to determine what should be sent on the optical links in the event that there is no data or control available to send. In such a case, either a Pad or and Idle will be sent. If there is currently a data packet being transmitted and the IL data FIFO has an under-run then a Pad will be generated and sent. Otherwise, there is not an active message and an Idle will be sent. For an input port link that is initially in the idle state, a high-to-low transition on the control marker indicates the start of a new message. Based on its determination of the VONIC-to-switch link state, this finite state machine also controls when words are pushed onto the FIFO. This constitutes the first half of the routing that takes place within the switch wherein data is fanned-out to one or more destinations. The fan-in selection of a single data packet to send back to the VONIC is performed in the Outward Logic. Data could simply be sent out on all optical output ports with the entire switching function performed by the Outward Logic. However, using a FIFO for each optical output is more efficient in that links that are

not needed are kept idle, less contention for control information that may need to be sent on other channels is created, and the arbitration in the Outward Logic is simplified. Using independent FIFOs also makes multicast operations more efficient by eliminating contention with control words that are not destined for the multicast set.

The remaining pieces of the Inward Logic including the FIFO buffer and transmit logic are instanced once for each optical output port on the chip as shown in Figure 7. One of the primary purposes of these two pieces is to resolve contention for the optical ports. Although there is only one switch port input connected to the Inward Logic, there is potential for some conflict due to flow control and other control characters that come from the OL block. For instance, throttle requests, nakBusy characters, and UR characters are handled by the Outward Logic, but the Inward Logic is responsible for transmitting the appropriate control words. Control information is given a higher priority than data within the switch, and so, data is buffered in the Inward Logic FIFO when necessary. A finite state machine within the transmit logic manages selecting between the multiple control character sources and the output of the data FIFO. Data and control characters to be transmitted on an optical output port are registered and sent through self-test and fault-tolerance blocks that will be discussed separately. The final element in the Inward Logic path is the VCSEL driver and VCSEL array where the digital voltage signal is converted to a current signal and then to a modulated light output.

3.2.3.2 Inward Logic Implementation

The Inward Logic for the switch core was implemented as a set of VHDL entities. The hierarchy of this code is shown in Figure 8. It is separated into pieces

corresponding approximately to the blocks within the block diagram with the two main pieces being the receive logic and the transmit logic. Some key inputs and outputs of this block are described in Table 5.

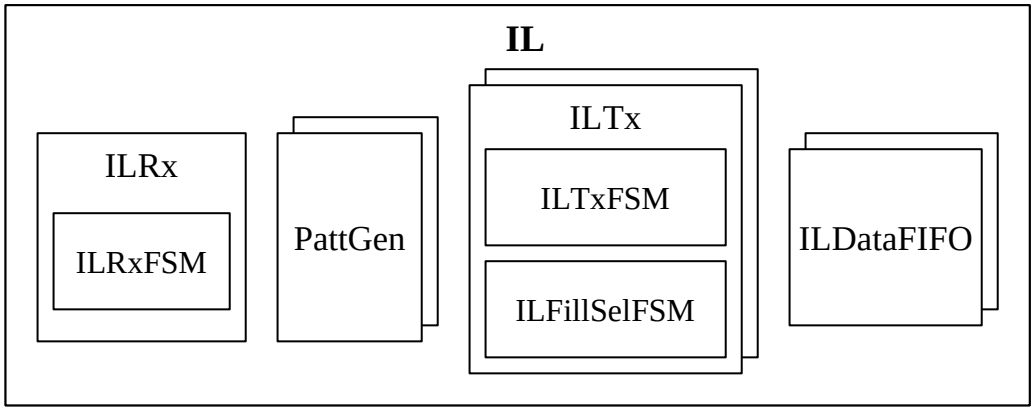

Figure 8. Hierarchy of Inward Logic VHDL code.

Table 5. Switch Inward Logic Ports

Name	**Direction**	**Size**	**Description**
datain	Input	32	Switch core input bus
ctrlBitIn_n	Input	1	Switch core input control marker
insertNakBusy	Input	1/port	Control request signal from Outward Logic
insertThrottle	Input	1/port	Control request signal from Outward Logic
insertUR	Input	1/port	Control request signal from Outward Logic
winner	Input	1/port	Arbitration winner from Outward Logic
insertPads	Output	1	Control request signal to Outward Logic
dataoutToTx	Output	32*ports	Data output ports to other switch chips
ctrlBitToTx_n	Output	1/port	Control marker output to other switch chips

The entity labeled ILRx implements the interface to the switch input port. The primary function of this logic is to monitor the input from the VONIC electrical link and decode control information in the data stream. It also provides routing information for these decoded control words in the form of a bit vector which gets sent to the ILTx entity to allocate access to the optical link. One bit of this vector goes to each of the ILTx entities that are associated with each optical output channel.

This ILTx code generates two key outputs. These are a set of multiplexor control lines and a word that is the protocol 3 encoded control word from the associated Outward Logic. The multiplexor control lines are used to control the switching between sending data, sending VONIC-generated control word(s), or sending the OL-generated control word that is encoded from this block. Logic within the ILRx entity is responsible for encoding the VONIC generated control words, since this will only need to be implemented once rather than once per switch port. Routing of OL generated control words is implicit because of the bit mask that the Outward Logic uses to communicate its request. Routing of VONIC generated control or data is external to this block.

The Inward Logic must make a selection between sending a data word out of the IL FIFO, an encoded control word from the ILRx, a fill word, an encoded control word from this ILTx entity, and an Idle character. This part of the Inward Logic creates a select line that is used to choose between control from the VONIC, control from the Outward Logic, and the FIFO output. If the FIFO is empty, a fill character is sent (either Pad or Idle) depending on the status of the ILRx finite-state machine. If none of these applies, then the optical link is idle and an Idle character is sent.

The PattGen entity is a part of the IL-transmit data path. It is implemented as two independent type-II (Galois) linear-feedback shift registers (LFSR) with matching tap points to create two 16-bit maximum length pseudo-random patterns. This type of LFSR was chosen for its maximum logic depth of one Exclusive-OR gate. This functionality is disabled during normal switch operation and adds very little overhead to the logic in terms of transistor count and delay, but it can be enabled in order to test the optical links.

3.2.3.3 Inward Logic Contention

Two control words cannot be received from the VONIC at the same time because there is only one input from the VONIC, but there is nothing to prevent the VONIC and the Outward Logic from issuing control commands at the same time. Issuing two control commands simultaneously from the Outward Logic does not benefit the overall system and therefore it is precluded from doing so. This means that the only source of control contention that needs to be considered is between VONIC-generated control commands and OL-generated control commands. The possible conflicts are enumerated in Table 6.

Table 6. Sources of Inward Logic Control Contention

VONIC	Outward Logic	Conditions
Throttle	insertThrottle	Happens when VONIC FIFO and OL FIFO are full at the same time.
Throttle	insertUR	Low probability – A node knows which port it is connected to before receiving data and so would not need to request a UR response.
Throttle	insertNakBusy	Low probability - contention would occur if sending a nakBusy to a node which was already being listened to (and also being throttled)
Pad	insertThrottle	Happens with bi-directional communication between 2 nodes: The OL FIFO of one fills and at the same time, the other node requests Pads. Ex: A→B. B→A. B's VONIC Throttles. So, A is sending Pads. B is still sending data to A when A's OL FIFO fills. A's OL sends insertThrottle to A's IL.
Pad	insertUR	Low probability – contention occurs if one host is sending to another host and while receiving, that second host request a UR packet to be sent to it.
Pad	insertNakBusy	Happens when a node is sending and receiving simultaneously and receiving a new connection while it is throttling its outgoing data stream. Ex: A→B. C→A. B requests a Throttle, so A sends Pads to B. B tries to start a transmission to A. A needs to send nakBusy to B (at same time sending Pads to B)
nakError	insertThrottle	Happens when FIFO fills on subsequent transmission to same destination. Ex: A→B, finishes and sends again. B's OL FIFO fills and issues insertThrottle to B's IL. B's VONIC sends nakError (based on earlier message).
nakError	insertUR	Low probability – Happens if a node request a UR after sending a message (received with errors).
nakError	insertNakBusy	Happens when a message was received in error and a new message transmission from another node has already begun. Ex: A→B. Finishes. C→B. A tries to send to B again. At same time B's VONIC sends nakError to A and B's OL sends insertNakBusy to A.
WhoAmI	insertThrottle	Low probability – assuming that the WhoAmI/UR arbitration is done at power on and not repeated later. Host should not send data before it knows who it is.
WhoAmI	insertUR	
WhoAmI	insertNakBusy	

As shown in the table, many of the possible control contention situations would not occur in a properly functioning system. The situations that are more problematic and of more relevance will be discussed here in order to explain how all of these situations are handled.

One solution to the problem of control contention is to combine control messages within the switch core. For the case of contention between Throttle and insertThrottle combining the two results in one taking precedence over the other. If the Throttle from the Outward Logic takes precedence, this means that a fixed number of Pads will be requested and the OL FIFO will receive relief after one round trip delay to the sending VONIC. (The fixed number of Pads is setup at power on.) This delay is taken into account in the FIFO-Full threshold level. This will not relieve the receiving VONIC at all however, because the OL FIFO is draining into it. The VONIC FIFO would only get relief if the OL FIFO actually had an under-run as a result of the throttle request. If the VONIC throttle request takes precedence, the recipient Outward Logic FIFO will continue to fill while Pads are being sent from this Outward Logic to the recipient VONIC. The recipient OL FIFO will stop filling after the Throttle from the VONIC reaches the sending VONIC and it begins sending Pads. This does not, however, help empty this FIFO as it is receiving no data but also sending no data. When the requisite number of Pads has been sent to the receiving VONIC, data will again be coming into and going out of the recipient switch Outward Logic at the same rate. Thus this FIFO will be no less full than before the Throttle. The situation of the recipient VONIC's Throttle causing the switch Outward Logic FIFO to fill faster because it cannot send data out, but continues to receive data can be mitigated by appropriately setting the Almost-Full threshold of this FIFO.

In addition to the Throttle contention just discussed, other control contention exists where the best option is to buffer the control messages and thus control contention is resolved by storing control messages and sending them sequentially. The three possible scenarios for this type of control contention are considered next.

In scenario 1, the VONIC sends a Throttle first. That is, for a given Host B that is receiving a message, the VONIC-B FIFO fills and a Throttle is sent to the switch-B Inward Logic, which in turn sends an insertPads signal to the switch-B Outward Logic and a Throttle to the transmitting switch Outward Logic. Next, the initial Throttle was received at the switch-B Inward Logic, and an insertThrottle was received from the switch-B Outward Logic due to its FIFO getting full. This situation is not really control contention because the insertThrottle (assuming no other contention) can be sent to the transmitting switch Outward Logic. When the resulting Throttle reaches the transmitting VONIC, it will already be sending Pads due to the initial Throttle. If it adds the Pads requested by the new Throttle to those remaining, the switch-B Outward Logic Pad count will expire, it will again begin sending data to VONIC-B Outward Logic, and it will continue to receive Pads from the transmitting switch Inward Logic. Thus both FIFOs are relieved.

In the second scenario, the Outward Logic throttles first. Again Host B is receiving a message. This time the SWIC-B Outward Logic FIFO fills and sends an insertThrottle to the switch-B Inward Logic. The switch-B Inward Logic responds by sending a Throttle to the transmitting switch Outward Logic, which forwards it to the transmitting VONIC, which, in turn, begins sending Pads. Immediately following the throttle request from the switch-B Outward Logic, the VONIC-B Inward Logic sends

a Throttle to the switch-B Inward Logic. Again, there is not explicit control contention, but there is an issue with the insertPads signal that would usually be sent to the switch-B Outward Logic at the same time as the Throttle is forwarded to the transmitting switch Outward Logic. The insertPads is still sent as usual so that this timing of events does not require a special case. The insertPads has the usual effect and the switch-B Outward Logic FIFO stops emptying. The transmitting VONIC again adds the requested Pads and continues to send Pads until all requested Pads have been sent. Again, the switch-B Pad countdown will expire before it stops receiving Pads and the switch-B FIFO then continues to empty.

The third Scenario is the case of simultaneous Throttles. This time Host B is receiving a message and gets a Throttle from the VONIC-B Inward Logic and the switch-B Outward Logic at the same time. This is a control contention because the switch-B Inward Logic cannot send both requests at once. One throttle request will need to be sent after the other. With the buffering of control characters this situation is converted into one of the previous two scenarios and handled appropriately.

The complimentary control signal to the Throttle character, the Pad character, is a bit more complicated. This is because there are multiple logic blocks which are potentially distributed within the switch and network that each are tracking or impact the number of Pads that have been sent in response to a Throttle. In order to avoid problems of contention with Pad characters, which are considered by the network to be control characters, some special handling of Pad characters is designed into the switch core. The ultimate purpose of the Pad character is to keep a packet traveling through the network intact in the event that there is a buffer availability problem downstream. It does this by keeping the link active since the end of a packet

is determined by the link going idle, while at the same time allowing a buffer which is approaching capacity to be emptied. To simplify the problem of control contention with Pad characters and to improve the efficiency of the switch core, Pads are created and discarded in multiple places within the network as needed. For instance, if a Pad character needs to be preempted to due to a insertUR, insertThrottle, or insertNakBusy request, data transmission is also being preempted and therefore, the Pad character can be discarded but still counted as if it had been sent.

Based on the discussion above, it is clear that there must be some storage within the IL block for control words that have experienced contention. Deciding the amount of storage comes from examining the nature of the possible contenders. If two control words are received at once, one can be sent and the other stored and sent on the next cycle. If control words can be received each cycle, the problem of control contention and storage quickly escalates. Pads (a control word) will certainly be received from the VONIC for several cycles back-to-back. This means that the VONIC generated control words cannot arbitrarily have precedence over the switch OL-generated control words. As such, a Throttle cannot wait until all of the VONIC generated control (Pads) are sent. For the particular case of Pads coming from the VONIC, the simplification discussed above greatly reduces the burden of IL control character storage. Considering the frequency which other control words can be received and the fact that control characters destined to different ports can be handled independently leads to fairly limited storage requirements. A finite state machine within the IL transmit logic is used to implement this storage which is effective in the remainder of the IL control contention situations.

3.2.4 Outward Logic

There are two main pieces to the Outward Logic. One is replicated once for each channel and monitors the data received from the associated Inward Logic (from another port). The second piece looks at what is decoded from each channel and from its own Inward Logic and controls what goes into the OL Data FIFO and what control messages need to be generated by its Inward Logic.

3.2.4.1 Outward Logic Block Diagram

Photodetectors and receiver circuits make up the start of the Outward Logic data path. These are followed by fault-tolerance and self-test circuits which are the counterparts to those within the Inward Logic data path. In the Outward Logic there is also a subdivision into receive logic and transmit logic, however their roles are swapped. The receive logic is associated with each optical input port and the single instance of the transmit logic is associated with the electrical output port. In addition to data and control FIFOs, the Outward Logic also has blocks to perform arbitration and control resolution. This data path ends with a boundary-scan enabled electrical output port, which is then connected to a particular host's network interface card. The Outward Logic data path is illustrated in Figure 9.

Receive logic connected to each optical input port monitors the incoming data in order to detect specific situations. First, it detects whether the incoming link is idle by examining the control bit marker and data input. This information is used for arbitration. In protocol 2, data packets are isolated from one another by a single Idle character without any other framing information.

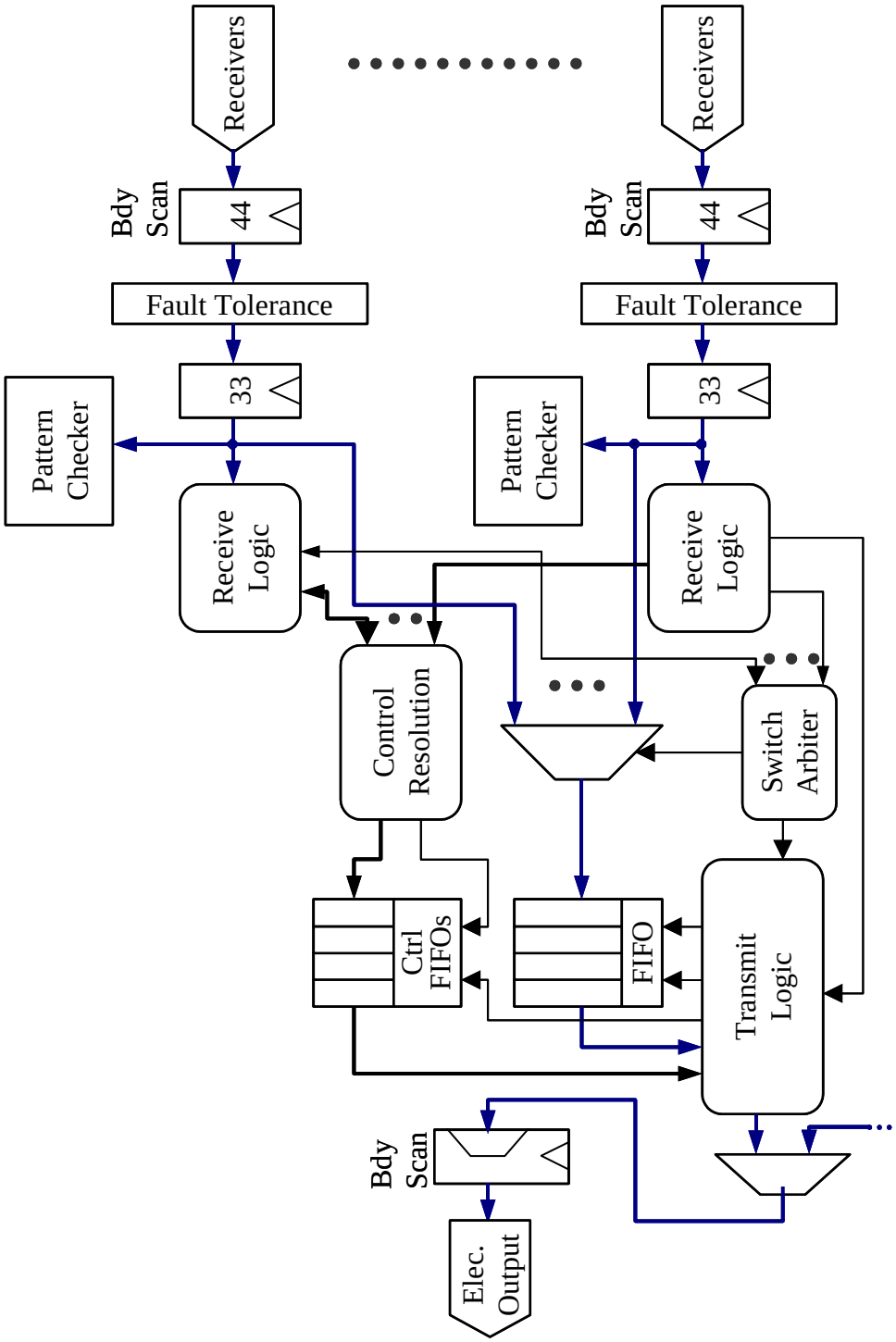

Figure 9. Outward Logic block diagram.

Within the switch this Idle character is converted into a protocol-3 start-of-message (SOM) character. The receive logic monitors an idle link for this character in order to determine when arbitration for the outward link is required. When the link is active, it monitors the link for the management character Throttle. Receipt of a Throttle is indicated to the transmit logic of the Outward Logic block. nakBusy, nakError, and UR characters are also monitored by the receive logic.

The receive logic portion of the Outward Logic also generates signals which are sent to the Inward Logic of the same Switch ASIC. This is done when control information needs to be communicated to other chips within the switch because the ability to communicate with these chips is controlled by the Inward Logic. If a Who-Am-I character is received, then a signal is sent to the Inward Logic to force it to interrupt any data message to send a one-word UR character. The particular UR character to send is based on which instance of the receive logic requested it. The current arbitration winner information is passed to the receive logic in order for it to request nakBusy characters to be sent by the Inward Logic to a port which has begun a transmission for which there is output port contention. Finally, this block is responsible for forcing the Inward Logic to send a Throttle character to the arbitration winner in the event that the Outward Logic data FIFO is becoming full.

The transmit side of the Outward Logic controls when data and control are taken out of their respective FIFO buffers and placed at the electrical output port. The following list indicates the data and control packets which are placed on the outward electrical port by the transmit logic in order of highest to lowest priority.

Throttle - bypasses Outward Logic Data FIFO

nakBusy - bypasses Outward Logic Data FIFO, sent from Outward Logic of a message recipient via its Inward Logic to losers of arbitration

nakError - CRC (or other) error, originates in recipient VONIC

Pad - sent when a message is in progress, but no Throttle or under-run

UR - sent as response to a Who-Am-I request

Data - output of Outward Logic Data FIFO

Idle - K28.5 character

The control FIFO shown in Figure 9 is made up of several smaller FIFOs corresponding to different control situations. Selection between these control FIFOs and the data FIFO is made by a finite state machine within the transmit logic by controlling when the individual FIFOs are read. Based on inputs from the receive logic and the output of the arbitration unit, the transmit logic also controls when data is pushed onto the data FIFO. The actual selection between data from the different optical ports to be pushed onto the data FIFO is made by the arbitration unit. A simple arbitration scheme wherein the port with the lowest port number receives priority in cases of contention has been implemented. This scheme is not “fair” but developing or implementing such an arbitration algorithm is beyond the scope of this project. The lack of fairness in the arbiter does not affect the handling of control packets within the switch because they are not subject to this arbitration, but rather have priority over data packets. Control or data packets which have been selected for transmission on the electrical output port by the transmit logic are first registered using boundary scan flip-flops and then sent out.

3.2.4.2 Outward Logic Implementation

The Outward Logic of the switch core is implemented as a set of entities in VHDL. These are illustrated in Figure 10 and closely follow the division of the logic given in the block diagram above. This hierarchy was chosen to allow synthesis scripts to optimize finite state machines independently and to provide modularity for some components such as the arbiter, which were changed during the design process. The primary inputs and outputs of the Outward Logic block are summarized in Table 7.

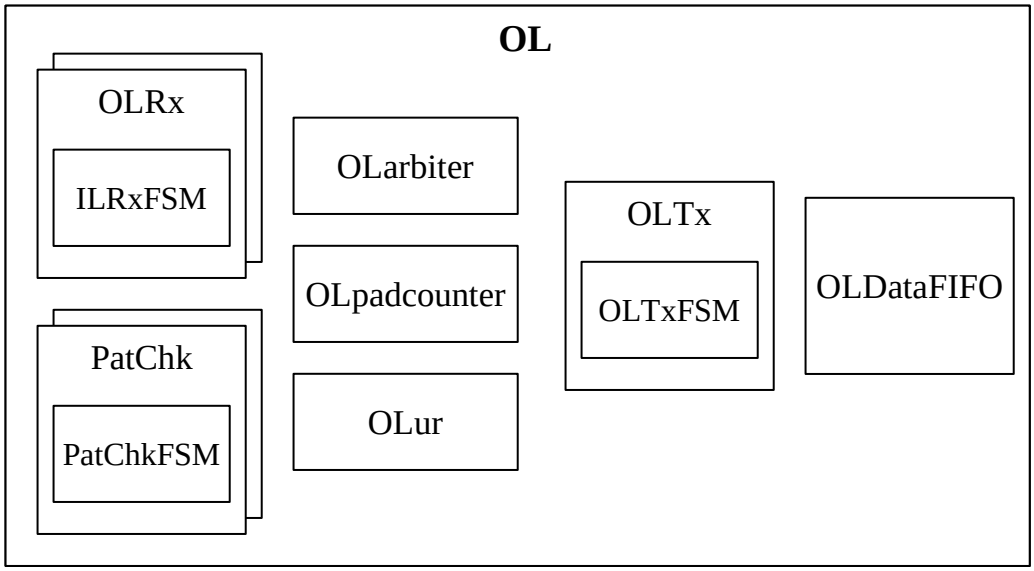

Figure 10. Hierarchy of Outward Logic VHDL code.

Table 7. Switch Outward Logic Ports

Name	Direction	Size	Description
datainFromRx	Input	32*ports	Data input ports from other switch chips
ctrlBitFromRx_n	Input	1/port	Control marker input from other switch chips
insertPads	Input	1	Control request signal to Outward Logic
padsToInsert	Input	8	Decoded number of Pads to insert
dataOutput	Input	32*ports	Data output ports to other switch chips
ctrlBitOut_n	Input	1/port	Switch core input control marker
insertNakBusy	Output	1/port	Control request signal from Outward Logic
insertThrottle	Output	1/port	Control request signal from Outward Logic
insertUR	Output	1/port	Control request signal from Outward Logic
winner	Output	1/port	Arbitration winner from Outward Logic

3.2.4.3 Outward Logic Contention

There is much greater potential for contention in the Outward Logic. At any given time, there can be as many control messages received as there are ports. This is the standard case of output port contention experienced by all switches. However, a goal of this switch design is to alleviate such contention among control packets. Contention between multiple data packets destined for the same output port is handled in the switch core insofar as it generates nakBusy characters to send to the sources of packets that are being dropped. Output port contention for data packets is handled as a line-card function as is common in commercial switches.

Table 8. Sources of Outward Logic Control Contention

Control	Control	Condition
UR	Throttle	Happens when WhoAmI sent when transmitting data
UR	Pad	Happens when WhoAmI sent receiving a throttled message
UR	nakBusy	Happens when WhoAmI sent when transmitting data
UR	nakError	Happens when WhoAmI sent when transmitting data
Throttle	Pad	Happens with inward and outward connections at one Host
Throttle	nakBusy	Happens with multicast messages
Throttle	nakError	Happens with sequential messages to same destination
nakBusy	Pad	Happens with inward and outward connections at one Host
nakBusy	nakError	Happens with sequential messages to same destination
nakError	Pad	Happens with inward and outward connections at one Host

Mitigation strategies for OL control contention include the following. Negative-acknowledgements are merged in the Outward Logic before sending them to the VONIC. Multiple Throttles are merged in the Outward Logic by taking the largest throttle request and sending it to the VONIC. Additionally, a throttle hysteresis is implemented which prevents the switch from becoming over-run with throttle requests. Pads are not forwarded from the receivers to the VONIC up-link. Instead, these Pads are discarded at the OL and Pads are inserted by the OL only at appropriate times. This eliminates the problem of multiple Pad contention, but also keeps Pads from interrupting other control or data. Idle status is determined by the Outward Logic and sent when the VONIC up-link is idle. This is not directly based on input data links being idle and so Idle characters do not compete with other control characters. The WhoAmI character is not ever sent on the VONIC up-link, but rather is handled and terminates within the switch. The UR character is sent on VONIC up-link instead and UR packets are arbitrated and intentionally limited by the Outward Logic.

Implementing these control contention strategies results in the possible contention being reduced from all combinations of the situations listed in Table 8 to the possibility of needing to send four control words simultaneously. That is, at a given instant, the output port of the switch may need to send one nakBusy, one nakError, one Throttle, and one UR. This more manageable situation is resolved by including a small OL control FIFO. Because of the priority given to control packets by the network protocol, this buffer is emptied before the OL data FIFO, which combined with the delay between possible control packets, limits the probability that this control FIFO will over-run.

3.2.5 Testability, fault tolerance, and configuration

There are a number of other connections and logic blocks shown in Figure 7 and Figure 9 as well as other configuration and status registers. These are used for setting up the Switch ASIC for normal operation as well as to implement testability features. Both the Inward Logic and Outward Logic can be bypassed for testing purposes. In addition to the ability to test optical links by using the boundary scan logic included at the transmitter inputs and the receiver outputs; the Switch ASIC has a built-in self-test (BIST) functionality. This consists of pseudo-random pattern generators at the transmit side of the Inward Logic and pattern checkers at the receive side of the Inward Logic. These can be enabled on a per optical port basis and will result in pseudo random data being sent over the optical link and checked for errors upon receipt. Bit errors and word errors are tracked in the pattern checker such that bit error rate can be observed. Making small adjustments to the existing datapath registers to produce the pattern generators reduces the additional logic needed for implementing this BIST functionality.

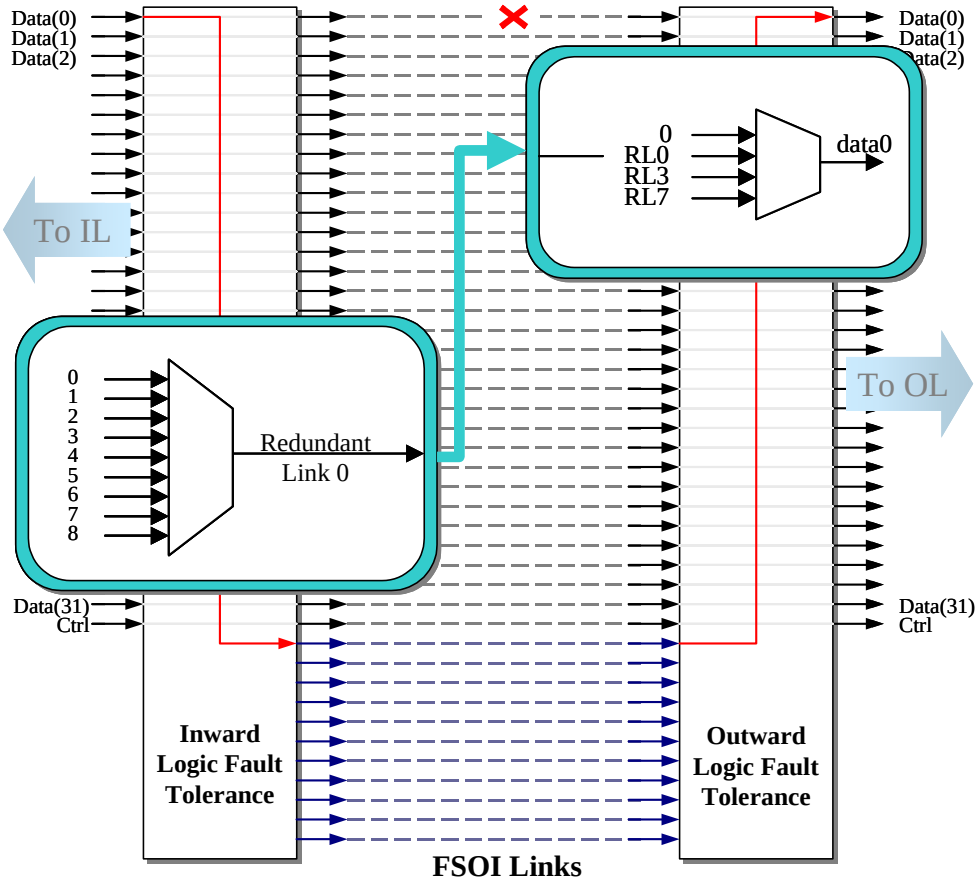

Figure 11. Optical link fault recovery strategy. Optical link for Data(0) is shown as faulty with redundant link(0) substituting for it. Input/Output registers, transmitters, receivers, etc. are not shown for the two Switch ASICs represented on the left and right here.

These BIST features, like the Inward and Outward Logic data paths, are behind fault tolerance modules that compensate for non-functional optical links. The optical ports are actually 44 bits rather than just the 33 bits that make up the data word and control marker. This allows for 25% redundancy in the optical links. While this seems high, it was chosen as a risk reduction method based on reasonable optoelectronic device cluster size and allows for a faster implementation of fault

tolerance to be used. Full fault tolerance would allow for any 11 of the 44 links within an optical port to be faulty. However, this requires a 33-to-1 multiplexor at each of the redundant link inputs on the Inward Logic side and a 13-to-1 multiplexor at every input on the Outward Logic side. This overhead leads to increased area and has a large impact on operating speed. As an alternative, a reduced fault tolerance scheme was developed and implemented. Under this scheme, each of the 33 optical links needed is backed up by three redundant links. Therefore, each of the 11 redundant links is a backup for nine normal links. By carefully assigning which redundant links serve as backups for which normal links, in the worst-case, faults on five normal links within a single redundancy group can be accommodated. This method reduces the logic to 33 4-to-1 multiplexors on the Outward Logic side and 11 9-to-1 multiplexors on the Inward Logic side. This is illustrated for one redundancy group in Figure 11.

There are many registers within the switch core that hold setup settings for the chip, such as VCSEL drive strength settings, and registers which can be read in order to get diagnostic information out of the Switch ASIC. These are accessed by way of several scan chains. An IEEE 1149.1 or JTAG (Joint Test Action Group) interface is used to read from and write to them [42]. These scan chains and associated registers are summarized in Table 9. In this table the size of each register is given in terms of the parameters NumPorts, DataWidth, and CharWidth. For the implemented design these values are: 9 ports, 32-bit data words, and 8-bit characters as reflected in the *size* column.

Table 9. Switch Core Externally Accessible Registers

Chain	Register	Description	Generic Size	Size
Registers Written by Switch Core				
1	olFifoFull	Outward Logic FIFO overflow flags	5	5
	ilFifoFull	Inward Logic FIFO (1 per port) overflow flags	NumPorts	9
2	lfsrLock	Pattern Checker synchronization status flags	NumPorts	9
3	bitErr	Number of individual bit errors detected by Pattern Checker	2 * Datawidth * NumPorts	576
	wordErr	Number of word errors detected by Pattern Checker	14 * NumPorts	126
	wordErrTc	Word error counter overflow flags	NumPorts	9
Registers Read by Switch Core				
4	olFtConfig	Outward Logic fault tolerance mux configuration	64 * NumPorts	576
	ilFtConfig	Inward Logic fault tolerance mux configuration	40 * NumPorts	360
5	fixedThrottleSize	Number of Pads to insert for Outward Logic requested Throttles	CharWdith	8
	olThrottleTimeoutMax	Throttle hysteresis setting	4	4
6	vcselConfig	Bias and modulation power level settings	352 * NumPorts	3,168
7	rxEnable	Receiver enable settings	44 * NumPorts	396
8	runLfsr	Pattern Generator mode control setting	NumPorts	9
	runCheck	Pattern Checker mode control setting	NumPorts	9

The functionality of the switch core logic described above has been verified by extensive simulation. Test cases were developed to specifically target circumstances to confirm the response of the switch logic.

3.3 Comparison with Conventional Switch Implementations

The VIVACE switch design was carried out independently of commercial protocols and standards, but utilized concepts common within the industry to build a new network protocol and switch implementation. The target application for this switch was for use within workstation clusters in order to speed up distributed computations by reducing the inter-processor communication time using high-bandwidth links between hosts and the switch and switching these links with low-latency. At design time, there were no commercial products available to fill this need. Many computer clusters have traditionally used 10 Mbit or 100 Mbit Ethernet to connect multiple hosts. Other high-performance clusters have used Myrinet networks for this purpose. The VIVACE design sought to avoid the overhead associated with Ethernet and provides higher bandwidth than available with Myrinet by developing custom communications protocols and using 10-gigabit links. The commercial need for such technology is evidenced by the development of Gigabit Ethernet and 10 Gigabit Ethernet standards and new Myrinet products as well as 10 gigabit switched interconnect for inside-the-box communication.

The architecture presented here uses space division multiplexing to make connections across the switch fabric. The physical implantation of this switch fabric will be discussed later, but it resembles more closely the “pizza-box” type of switch as apposed to a chassis-based system. This stems from the wide electrical paths that are used to bring data into the single switch fabric component (the MCM). Scaling the

number of ports on the switch and adding more network/traffic management features would change the physical implementation to one more resembling a chassis mounted commercial switch. The switching scheme presented here uses packet switching with variable length packets. This allows for very low overhead for applications transferring large blocks of data, but requires policing functionality (at the application level in this case) to ensure that ports are not dominated by a single sender. The data buffering with in the switch core is not intended to implement a store-and-forward type architecture, but is rather to limit the possibility of data loss due to the transmission of control information which has a higher priority. Therefore, although there is potential for some variability of the latency through the switch fabric, it is very low. The multiplexor-based fabric with full connectivity presented here leads to inherently deterministic routing.

The switch design presented here is a custom design that meets the goals of the VIVACE program in providing low message overhead, low latency, and high bandwidth interconnection between a cluster of workstations. The design process took advantage of existing switch features in implementing an architecture that exploits the full connectivity among the multiple switch chips making up the VIVACE switch fabric. Mapping this architecture into a CMOS IC design for free-space optical interconnects will be discussed in the next chapter.

Chapter 4

FSOI BACKGROUND AND USE IN THE VIVACE SWITCH

Free-Space Optical Interconnect (FSOI) is a technology that replaces electrical signaling with unguided-wave optical signaling to provide communication between two points. This technology has many advantages which are particularly useful in applications where high-density interconnect between multiple points is desired. Hybrid integration of optoelectronic devices to CMOS VLSI circuits provides the opportunity to design ICs that integrate millions of transistors and thousands of high-speed optical I/Os for high-performance computing and switching applications. For the VIVACE program, the rich interconnection among switch chips on a multi-chip module required to implement a fully connected crossbar fabric is made possible by using free-space links to realize the inter-chip connections.

This chapter will provide some background information on optoelectronic devices and how they are used in the architecting of the packet switch. It will also cover details of incorporating free-space optical inputs and outputs in standard digital designs. Further details of the circuits used and the physical characteristics of the VIVACE hardware will be presented in the following chapter.

4.1 Optoelectronic Components

Free-space optical links are created by combining light emitting devices with photon detection devices by using optic elements to couple the light from the emitter to the detector. A number of devices can be used to create such links. An

emitter can be a device that creates optical power itself or one that can modulate another optical source. Light emitting diodes and laser diodes are common semiconductor devices that convert electrical power into optical power and serve as emitters in many optical systems. Vertical-Cavity Surface-Emitting Lasers, or VCSELs, are one type of laser diode that is particularly well suited to use in free-space optical links. Another device that shows great promise in creating FSOI-based systems is spatial light modulator. These devices can affect the propagation of incident light based on an electrical input and have the additional benefit that they can be used as photon detectors. Other devices that are used as detectors include avalanche photo diodes, metal-semiconductor-metal (MSM) diodes, and p-i-n diodes. In order to produce semiconductor emitter and detector devices with more desirable properties, a variety of fabrication materials have been used. This tends to lead to separate fabrication processes for the electronic circuitry and the optoelectronic devices. The VIVACE system makes use of VCSELs and p-i-n diodes fabricated on a gallium-arsenide substrate, which will be discussed briefly next.

4.1.1 VCSELS

As the name suggests, VCSELs are laser devices that have their optical cavity oriented vertically and, as such, emit light normal to the surface of the wafer in which they are fabricated. Although it may at first seem trivial, this vertical orientation leads to many of the desirable features of the VCSEL. In contrast to traditional edge-emitting semiconductor laser devices, VCSELs are well suited to array processing. This is useful in batch processing, allowing mass-production and wafer-level testability. It also allows two-dimensional arrays to be created to giving dense optical interconnect possibilities. Other VCSEL characteristics that make them

useful in FSOI systems are their narrow beam divergence and low threshold current [43]. A good historical perspective of the development of the VCSEL device is presented in [44]. The development of the VCSELs used in the VIVACE program is described in [45][46].

VCSELs are typically made to lase by creating a large current density in the cavity; however, due to their small size currents on the order of only milliamperes are required. Two electrical contacts are required for each device, although in some VCSEL array products one contact is common among all devices in the array. Some characteristics of interest in designing optical links based on VCSELs are the aperture size, the threshold current and the slope efficiency. The VCSELs used in the VIVACE program have a 6-μm aperture, a threshold current of 0.8 mA to 0.9 mA and a slope efficiency of approximately 0.5 mW/mA.

4.1.2 Photodetectors

The principle behind most photodetectors is the creation of a space-charge region by a pn-junction wherein the electron-hole pairs created by incident photons can be swept out and detected as a photocurrent. Greater efficiency is achieved in the p-i-n diode, which is the type of device used in the VIVACE program, by creating a larger photon collection area with an intrinsic semiconductor region between the p-type and n-type regions of the diode.

The VIVACE photodetectors have a 60-μm diameter, a responsivity of 0.5A/W (for incident light with a wavelength of 850 nm), and a dark current 5nA (with 3-volts reverse bias).

4.2 Interface Electronics

The interface between the optoelectronic devices and digital CMOS VLSI circuits is accomplished using VCSEL driver and optical receiver circuits. VCSEL driver circuits work by translating a digital logic-level signal into a current-mode output signal, which is used to modulate the VCSEL. The optical receiver circuit converts weak current signals at the photodetector into logic-level signals that are fed to CMOS logic for processing. An optical receiver design must balance power consumption, operating speed, bit-error rate, rejection of crosstalk and power-supply noise, and chip area. Fiber-optic communication links are now very prevalent and the receiver circuits used with them have been widely studied [47][48]. As noted in [49], the design of receiver circuits for use in systems of two-dimensional optoelectronic device arrays poses unique challenges. Their physical size and power consumption must be constrained relative to the receivers used in telecommunications systems. This must be done while limiting the optical power required and maintaining acceptable operating speed.

4.2.1 VCSEL Drivers

The VCSEL driver circuit converts a CMOS digital signal into a current-mode signal suitable for driving a VCSEL device. In general, this means sourcing or sinking two current levels; a bias current and a modulation current The bias current is used to provide a constant current to the VCSEL in order to set is operating point relative to its threshold. The modulation current provides the additional current necessary to increase the light output of the VCSEL such that the change in light output can be detected in order to receive optically transmitted data. A number of VCSEL driver circuit configurations are possible. The approach that has been used

here is based on the simple and robust circuit configuration shown in Figure 12 [50]. It consists of two PMOS drain-shorted transistors that are designed to source current while operating in saturation. The circuit operates on the principle that a MOS transistor in saturation acts as a current source. It consists of two drain-shorted transistors (M1 and M2) that supply the modulation current and bias current, respectively. The drain of M2 is connected to the power supply rail. The gate of M2 is connected to an analog voltage that sets the bias current. The transistor M1 controls the modulation current. The gate of M1 is connected to a digital input signal. The source of M1 is connected to a power supply, VMOD, which controls the modulation current. The modulation current is directly proportional to the voltage VMOD.

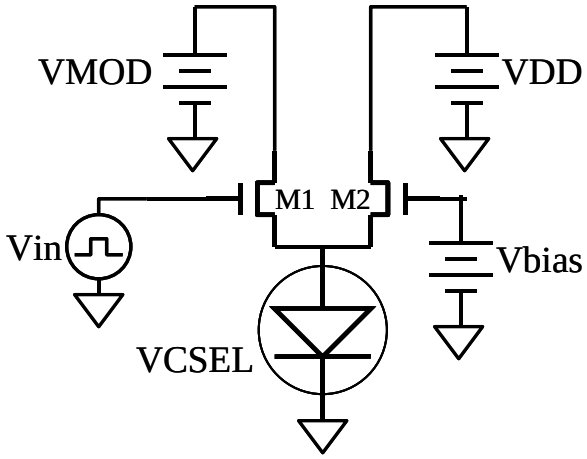

Figure 12. General VCSEL Driver cell.

4.2.2 Receivers

Receiver circuits are required in order to convert the current-mode outputs of photodetectors to voltage signals and amplify these signals to a level that can be used by the internal CMOS logic. Many circuit configurations have been used as receivers including inverting amplifiers, transimpedance amplifiers (TIA), and differential amplifiers [43][49]. The configuration used in VIVACE uses a TIA front end followed by further gain stages. The general architecture is shown in Figure 13 and is widely used in FSOI systems [51]. This circuit has the advantages of simple operation, low power consumption, compact area, and high-speed operation.

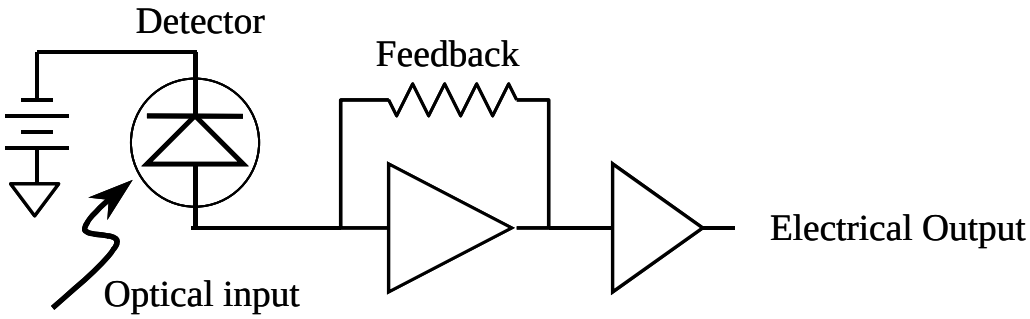

Figure 13. General Receiver architecture with TIA and buffer stages.

4.3 Design considerations

The use of free-space optical inputs and outputs within the design of a system takes special consideration throughout the design process. There are implications on the overall architecture, the physical design and the implementation.

In many cases this leads to additional complexity, however, significant advantages can be achieved, especially in terms of design scaling, which can offset this complexity.

4.3.1 Physical design and integration

One of the challenges in designing large-scale ICs for use in FSOI systems lies in the development of an efficient method for integrating existing VLSI circuit layouts with two-dimensional arrays of optoelectronic devices. Two reasons for this are the need to include fixed geometry and the need to communicate with non-CMOS level signals. Physical restrictions placed on the design to facilitate packaging and system-level use have to be planned for, and considered throughout the design process.

The dissimilar materials and processes used for optoelectronic devices and digital circuitry often leads to independent fabrication followed by physical integration of two (or more) distinct parts. This stems from the desirable optical properties of group III-V materials such as gallium arsenide and the maturity and ubiquitous availability of silicon processes for digital logic. In addition to this hybrid integration strategy, there are also efforts to perform monolithic integration of circuits and optoelectronic devices. Two monolithic integration approaches are fabrication of digital and driver/receiver circuitry in the III-V wafers used for the optoelectronic devices and the use of silicon photodetectors in standard CMOS processes. Research is also being done on making efficient light emitting structures in silicon.

An attractive method for the hybrid integration of optoelectronic devices with silicon ICs is flip-chip bonding or bump bonding. In this process two die (or a die and package) are directly bonded together without the use of wire bonds by placing solder balls between pads on each surface and heating the assembly to allow

the solder to reflow onto both pads. This technology was originally pioneered by IBM Corporation in the 1960's under the name "Controlled Collapse Chip Connection", or "C4" (see, for instance, [52]). By distributing such pads in an array across the surface of the two chips comprising a hybrid device, much greater I/O bandwidth is available. This is made possible in free-space optically interconnected systems because there is not a need to electrically route out the connections to the pads within the array on a chip carrier or other substrate, which in turn, allows the pitch of the inputs and outputs within the array to be very small. The resulting devices, however, cannot be packaged in a traditional manner. For example, where traditionally two ICs might be packaged in separate plastic or ceramic packages and communicate via copper or gold traces across a printed circuit board, chips using FSOI must maintain line-of-sight interconnection pathways in order to communicate. They also require beam focusing and reflecting optics ranging in complexity from lenses and mirrors to external laser sources and computer-generated holograms to establish these links. These factors necessitate the development of custom packaging techniques in addition to the changes that must take place in the CMOS IC design process itself.

The number and location of optoelectronic communication ports must be considered. These are both largely determined by the optoelectronic devices that will be integrated. Fabrication capabilities limit the size and pitch of the optoelectronic devices, which in turn, limits the number that can be spread across a given area. The OE devices may be fabricated in a one- or two-dimensional array (or in discrete locations). Currently popular optoelectronic devices, such as Vertical-Cavity Surface-Emitting Lasers (VCSELs), Multiple Quantum Well (MQW) modulators, and Metal-Semiconductor-Metal (MSM) photodetectors, are two terminal devices and thus, two

contact pads must be associated with each optoelectronic device. One of these serves as the signal port of the OE device while the other is generally connected to a common bias voltage. Unless the optoelectronic devices share a common contact, this results in a regular array of contact-point pairs on the optoelectronic die which correspond to a necessarily matching array on the CMOS die. The size and shape of these pads is dictated by the bonding process to be used for the hybrid integration. Like conventional perimeter pads on a CMOS die, these area-distributed pads are formed from (minimally) top-level metal with a bond opening placed over it. Other physical design impacts of using this type of distributed I/O including computer design strategies and layout implications has been described in [53] and [54].

The VIVACE system uses a two-dimensional array of optoelectronic devices, which are then directly attached to silicon ICs by a bump bonding process. This array consists of monolithically integrated VCSELs and p-i-n photodetectors arranged in interleaved clusters.

4.3.2 Architectural effect

The primary architectural effect of using two-dimensional arrays of optoelectronic devices in the design of the VIVACE Switch ASIC is the availability of large bandwidth between each of the chips on the MCM. The connectivity for a single-MCM, nine-chip (3 x 3 array) switch implementation was previously shown (in Figure 3).

The increased bandwidth is illustrated as follows. The physical size of the optoelectronic device array used in VIVACE is 6.75 mm x 6.75 mm. In this area are the 396 VCSELs, and 396 photodetectors, which make up the subset of the full array that is available for chip-to-chip communication. The optical port density is,

therefore, 17.4 ports/mm^2. Considering the same size area and wirebond perimeter pads on a 75 μm pitch (the effective pitch of the staggered I/O pads used in VIVACE) the equivalent electrical I/O is 6.85 pads/mm^2. However, taking into account the need for power and ground pads, control pads, the use of differential electrical I/O, and the 25% redundancy budgeted to the optical links results in the nine-times higher optical data bandwidth than electrical data bandwidth available for the switch design.

The full-crossbar architecture is very well suited to this interconnection pattern because of its all-to-all connectivity among the distributed ICs. That is, each of the nine chips has a dedicated direct link to every chip. Such connectivity is desirable in switch fabrics because the resulting switch is non-blocking. In a strictly non-blocking switch, no configuration of existing connections can prevent a new connection from being established between an idle input port and an idle output port [55]. With a blocking network, contention within the fabric can prevent the connection of two ports that are not currently being used. The full connectivity also makes multicast and broadcast transmission inherently straightforward.

In addition to the connection pattern being well suited to switching applications, the switch structure maps very well into the interconnection pattern. Using two-dimensional arrays of optoelectronic devices as described above gives great optical bandwidth, but the electrical I/O which is located at the periphery of the chip is much more limited. In the switch design presented in Chapter 3, each switch chip has a single core that implements one input and one output port. Therefore, much less bandwidth (nine times less) is required for the electrical I/O.

4.3.3 Allocation of devices

The details of how the available optical and electrical I/Os are used within the VIVACE switch design will be given here. The total number of each comes from physical constraints in the system The optoelectronic device array fabrication process, the optical system design, and the hybrid integration and system assembly processes all contribute to the allocation of inputs and outputs, and ultimately, to the design of the switch chip.

The optoelectronic device array is a 36 x 36 array of VCSELs and p-i-n photodetectors, of which a subset are used. The subset consists of nine clusters, each with forty-four VCSELs and forty-four detectors. The devices are clustered in this way in order to match the lens system design and give rise to the forty-four bit paths that exist between each chip and every chip. This connectivity includes a link from each chip back to itself, which is a by-product of symmetry within the system design. The pitch of VCSELs and photodetectors is 175 microns in both the x- and y-direction, which is considerably tighter than the 250-micron pitch, which is common in linear fiber-optic ribbon cables.

The datapath from the electrical input of the switch through the logic and optical links to the output of the switch is thirty-two bits wide, with a thirty-third bit serving as a control flag. Allocating one of the forty-four optical links to each of these thirty-three bits leaves eleven links left over. These eleven links are allocated to fault tolerance redundancy to be used in the case of a non-functional link.

Available electrical input and output pads are dictated by the physical size of the ASIC, which is bound by the physical size of the optoelectronic device array and the space available on the MCM. Balancing these two constrains led to a resulting CMOS die size of 7.825 mm x 7.825 mm. The desired pitch of MCM traces,

to which the periphery pads would be wirebonded, resulted in 110 available electrical I/O (many of which are differential signals and including several analog bias inputs).

As a result of the device allocation described here, the electrical and optical data ports described in Chapter 3 were realized. In the next chapter the circuits used to implement these ports will be described in detail.

Chapter 5

HARDWARE DEMONSTRATION SYSTEM

A critical step in this research is validating the approach by demonstrating that the various technological challenges can be overcome in order to build a system capable of providing the free-space optical interconnect used in the switch architecture. To that end, a demonstration system capable of effecting, exercising, and characterizing optical connections between hybrid integrated circuits on a multi-chip module was constructed and tested. A primary concern in the development of components for this system was creating and testing wide optical paths between chips.

This chapter goes into the details of the electronic hardware developed for this system. Details of custom circuitry and ICs will be given along with test results. Printed circuit boards used in the system and test setups will be described along with descriptions of the multi-chip module, optomechanical system, and the assembly process. Finally, the overall system and test results will be presented.

5.1 ASIC development

During the VIVACE program four integrated circuit designs were launched in order to achieve the goals of the program. These designs were completed to varying degrees and led to the inclusion of one of the designs into the final system assembly and demonstration. These four chips will be referred to as the “Test ASIC”, the “Switch ASIC”, the “Transceiver ASIC”, and the “Interconnect ASIC”. The Test ASIC, and the Transceiver ASIC were designed, fabricated and tested. The Switch

ASIC logical design and implementation was completed, but further effort was placed on hold during the physical design phase. The Interconnect ASIC was designed and submitted for fabrication, but placed on hold prior to the start of fabrication due to successful test results from the Transceiver ASIC for which it was to be a backup.

A number of custom circuits were needed and developed for the VIVACE program. Among these were the VCSEL driver (also called Transmitter herein) and photodetector receiver circuits. In order to get early verification of these circuits (prior to the fabrication of the large-scale designs) the Test ASIC was fabricated and tested.

The Switch ASIC was conceived and designed to be the centerpiece of the VIVACE hardware demonstration in two regards. First, it was to interface with the large optoelectronic device arrays developed for VIVACE to provide free-space optical communication between multiple ICs on an MCM. Secondly, it was to perform the digital logic functionality required to build a multi-port, multi-gigabit per port packet-based switch capable of providing communication between several computer workstations.

The concept of the Transceiver ASIC came about as a program risk reduction. Its primary purpose was to meet the first goal of the Switch ASIC in order to demonstrate the large scale free-space interconnection achievable with the VIVACE design while reducing the chances of problems with the digital protocol or logic implementation jeopardizing the ability to demonstrate the optical interconnect.

The purpose and details of the Test ASIC and Transceiver ASIC will be discussed further in the following sections along with discussion and test results of the assembled Smart Pixel Array (SPA), which was based on the Transceiver ASIC.

5.1.1 Test ASIC

The VIVACE Test ASIC was a collection of circuits and sub-circuits designed and fabricated to provide characterization of the process to be used for the remaining VIVACE ASICs and to test, verify, and improve the custom circuits developed for the VIVACE program.

The silicon process chosen fairly early in the program for the fabrication of the CMOS ICs was the 0.25-µm process from Taiwan Semiconductor Manufacturing Company (TSMC) [56]. This process was selected based on the trade-off between fabrication cost and feature size as well as process maturity and availability of digital library cells. This process allows for 0.25 µm (drawn) gate lengths, one polysilicon layer, and five levels of metal interconnect. It is targeted for a core voltage of 2.5 volts with thick-oxide devices available for 3.3-volt input/output cells. Additionally, the mixed-mode variant of the process was chosen which adds the ability to make precision high-resistance polysilicon resistors, metal capacitors, and varactors that were used in the custom circuits. This mixed-mode extension to the standard logic process uses the same base process but is fabricated on non-epitaxial wafers.

The Test ASIC was fabricated as a part of a multi-project run in October 2001 and completed fabrication in January 2002. It was 4.374 mm x 2.174 mm (diced at 4 mm x 5.5 mm) and consisted of 112 perimeter wirebond pads and 92 probe pads. The circuits incorporated on the Test ASIC included the following:

- Multiple variants of the VCSEL driver circuit
- Multiple variants of the photodetector receiver circuit
- A custom LVDS driver

- A custom LVDS receiver including on-chip termination
- Break-out cells of critical pieces of these circuits
- Individual NMOS and PMOS transistor cells for characterization
- A custom library of Input/Output and power pads

5.1.1.1 VCSEL Driver

The interface to the VCSEL devices is a MOSFET-based current steering circuit that converts a digital data input into a modulated current signal. It provides both a constant bias current, which flows irrespective of the data input, and a modulation current, which is controlled by the data being transmitted. Each of these currents is controlled by a current-mode digital-to-analog converter (DAC) that is in turn controlled by a four-bit digital input and an analog bias voltage. An additional global bias voltage provides direct control over the current through the VCSELs. The purpose of the modulation current is the straightforward on-off keyed modulation of the laser output by the digital data input while the purpose of the bias current is to keep the VCSEL near its threshold when transmitting a logic low bit.

The VCSEL is operated at a current equal to the sum of the bias and modulation currents when transmitting a logic high bit. In operating VCSEL devices there is a trade-off between power consumption and achievable modulation rate based on the current at which they are biased. Naturally, increasing the bias current increases the power consumption, but biasing VCSELs near or above their threshold results in reduced turn-on time and a faster possible modulation rate. Inclusion of a digitally controlled bias current circuit within the transmitter circuit allows for both balancing the speed/power trade-off as well as for adjusting for non-uniformity across

large VCSEL arrays. Similarly for the modulation current control circuit, modulating the VCSELs at the lowest current level that yields the desired performance can conserve power.

The current-steering architecture used in the transmitter circuit comes from the system architecture. Since there are a large array of VCSELs being driven simultaneously, it is important to limit electrical crosstalk among the multiple channels. When a logic high bit is being transmitted a current, $I_{VCSEL\text{-}High} = I_{bias}+I_{mod}$ is being pulled through the VCSEL. The modulation current is steered to the other branch of the driver circuit when transmitting a logic low bit resulting in a current $I_{VCSEL\text{-}Low}=I_{bias}$ but a total current draw still equal to $I_{total} = I_{bias}+I_{mod}$. In this manner the transmitter circuit presents a relatively constant load to the power supply network on the chip and thus greatly limits the possibility of channel-to-channel crosstalk through the power rails. While this increases the overall power consumption of the chip it allows for reliable multi-channel VCSEL links.

Power for the VCSEL driver comes from a 3.3-volt supply rail. This same supply rail would also connect to the cathode of the VCSEL that is being driven as shown in Figure 14. This high voltage necessitates the use of thick-oxide devices to make the differential pair of the driver circuit, but it benefits the circuit by allowing for a voltage drop across the VCSEL while still providing adequate operating room.

The current DACs used to control the bias and modulation currents are constructed as shown in Figure 15. Each branch has transistor sizes twice the size of the branch preceding it such that the current increases linearly with the four-bit binary input word. The bias voltage for the DAC is generated on-chip from an input DC current reference and low-pass filtered within the transmitter circuit. This bias voltage

is shared by the bias current DAC and the modulation current DAC and controls the step size for each. The modulation current DAC is sized to provide twice the current of the bias DAC for a given digital setting. This provides for more precision at the lower current range needed for VCSEL biasing. A single-bit power-down capability was planned for the transmitter but not included because both DACs can be set to zero current, which effectively disables the transmitter output.

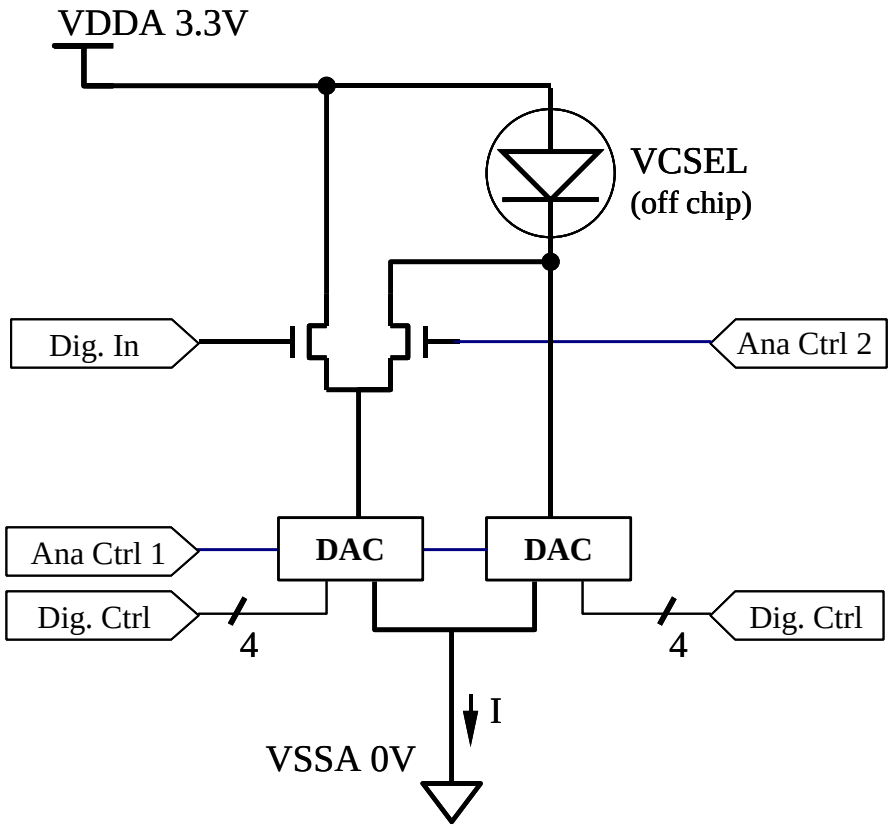

Figure 14. VCSEL driver circuit topology.

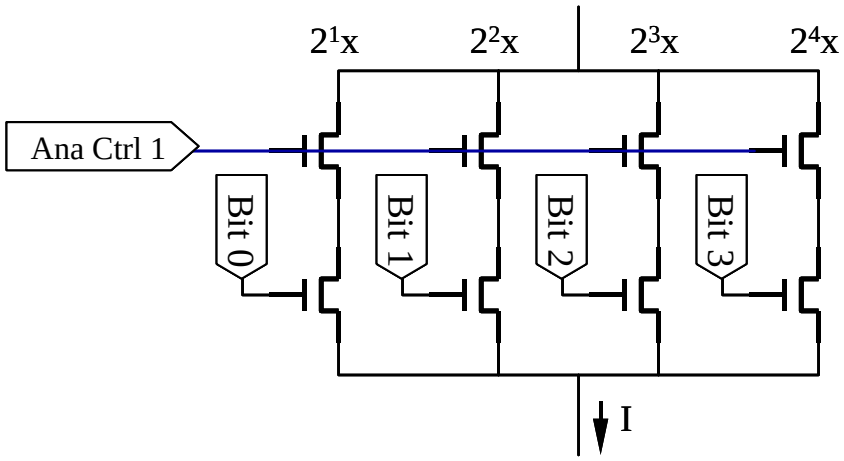

Figure 15. Digital to Analog Converter (DAC) circuit topology.

The effect to stepping through the Bias and Modulation settings using the DAC circuit is shown in the simulation results in Figure 16. The spikes in current output occur when multiple DAC control bits are switched simultaneously. However, these settings would be held constant during normal operation

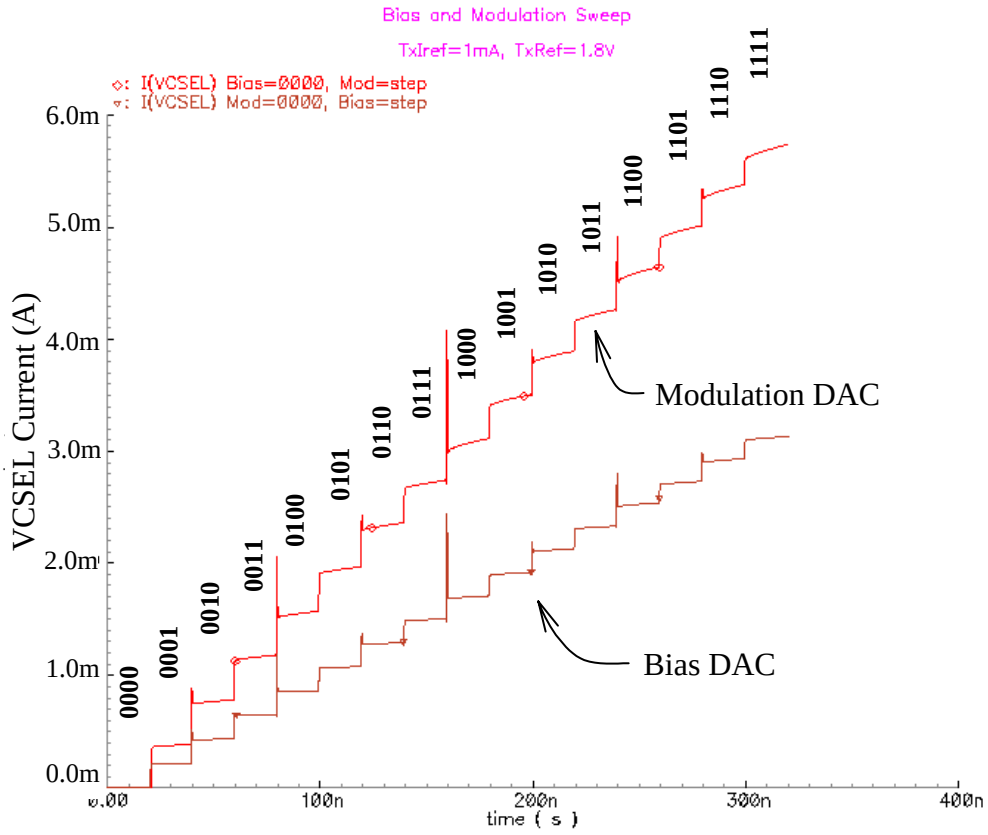

Figure 16. Transmitter DAC simulation. Digital control input is stepped from "0000" to "1111" for the bias DAC and modulation DAC independently. (The bias is held at zero for the modulation sweep and vice versa.)

5.1.1.2 Photodetector Receiver

Interfacing between the photocurrents produced in the detectors and the digital logic is done with the receiver circuit shown in Figure 17. This is a multi-stage amplifier consisting of a pre-amp, three differential post-amp stages, and a final CMOS buffer stage. The pre-amp stage is a transimpedance amplifier with variable gain, which converts the current input from the detector to a voltage signal that is fed

into the first post-amp. Three differently sized NMOS transistors in parallel make up the feedback network of the pre-amp. The gates of these feedback transistors are controlled by off-chip voltage sources to set the desired gain of the pre-amp. The gain of the post-amp stages is also globally adjustable by way of a voltage reference created on-chip from a DC current reference input. The post-amps and CMOS buffer (except for a final single-ended inverter) are implemented as differential circuits to improve noise immunity and reduce power supply switching noise. At the first post-amp stage a reference voltage is required to set the switching point. An additional, global, copy of the pre-amp circuit is used to generate this voltage from a DC input current source.

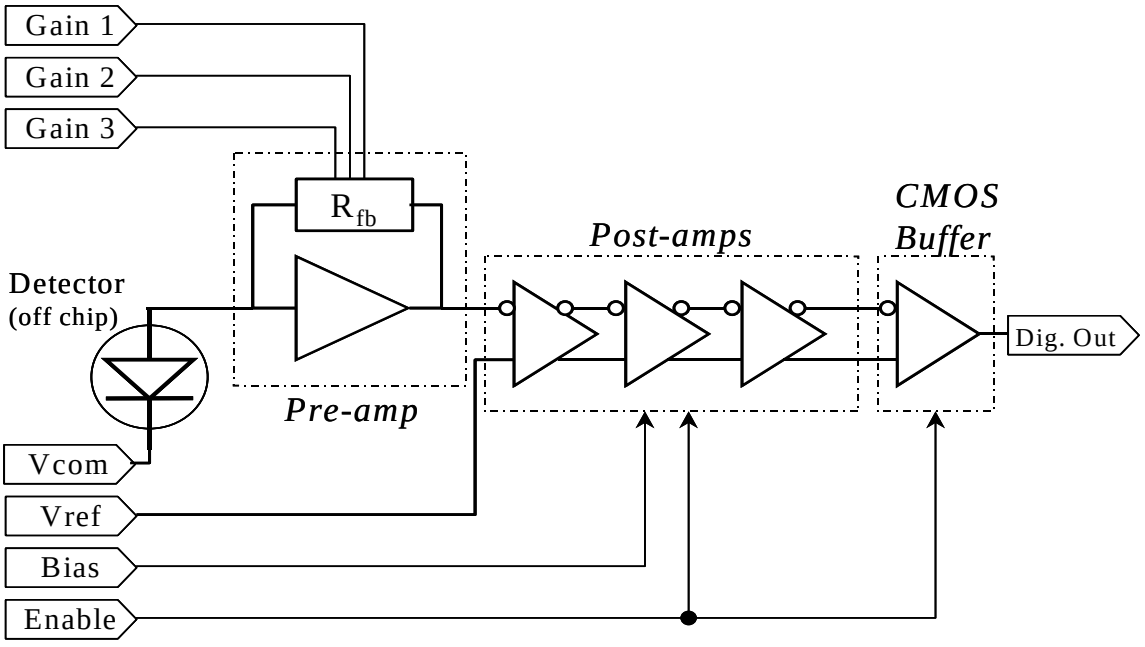

Figure 17. Photodetector receiver circuit topology.

An important feature of the receiver circuit is the ability to shut it down by way of an enable input that disconnects the post-amp stages and CMOS buffer stage from the power supply. This power-down functionality is included in the receiver as a testability enhancement as well as to reduce power consumption in the final system for links that may not be used. It also has the benefit of reducing switching noise generated from a receiver oscillating due to its input floating, for instance as a result of a non-functional detector.

5.1.1.3 Electrical Input/Output Buffers

An important decision was made regarding the final system architecture and electrical interface that gave further motivation to the Test ASIC fabrication and test. The target data rate of 10 Gbps for each switch port led to an electrical interface consisting of 32-bit wide buses clocked with a 300 MHz clock. This meant that the electrical I/O of the VIVACE ASICs needed to support both 300 MHz clock signals and 300 Mbps data. After careful consideration, modeling of the anticipated electrical environment of the MCM and Switch Motherboard, and simulation of the single-ended I/O cells available in the cell library for the TSMC process, it was decided that custom I/O cells would have to be developed and used in order to meet the target data rate. These cells developed were based on low-voltage differential signaling (LVDS) and thus required two bondpads per logical input or output. In the switch design the majority of the electrical I/O pads were to implement the input and output data buses using thirty-four LVDS input pairs and thirty-four LVDS output pairs. Therefore, it was efficient to develop a pad cell library based on the form factor of the LVDS input and output pads that were designed.

A significant role of the Test ASIC is the validation of the library of bondpad cells developed for use in the subsequent VIVACE chip designs. This library consists of the following pad cells:

- LVDS Input pad pair
- LVDS Output pad pair
- CMOS Input buffered pad
- CMOS Output buffered pad
- Analog I/O pad
- 3.3V Power pad
- 2.5V Power pad
- 0V Ground pad

The LVDS input circuit consists of a differential receiver with an on-chip 100-ohm line-to-line termination resistor and a CMOS buffer stage. The circuit is designed be compatible with commercial LVDS output drivers. It operates with a nominal common-mode input voltage of 1.15 V and differential-mode voltage swing of 300 mV. Excluding power dissipated in the termination resistor, this circuit dissipates 0.5 mW from a 2.5-volt supply. It is self-biasing and requires no control or bias inputs. Additionally, the on-chip termination reduces component count on the MCM and improves signal integrity.

For the high-speed outputs, an LVDS output driver circuit is used. This is again a fully differential CMOS driver, which is designed to drive off-chip interconnect with 100-ohm differential impedance and a line-to-line termination. While this methodology impacts the VIVACE system design in that the SPA pin count is doubled for the same number if outputs, it helps solve problems of simultaneous

switching noise and increases noise immunity. The circuit takes a single-ended, 2.5-volt digital input signal and creates a true/compliment signal in the first stage. Two internal stages condition this signal for the final LVDS driver stage. This final stage employs a push-pull topology, which based on the bias voltage input, drives a variable current to the load. The LVDS output can be disabled by way of a digital output enable signal that shuts down the CML buffer stages and the LVDS output stage.

The remainder of the pad cells developed for the Test ASIC are relatively straightforward and include CMOS I/O pads, bias-voltage generating pads and power pads that separately power the LVDS circuits, VCSEL driver circuits, receiver circuits, core digital logic, and CMOS I/O pads. Several of the bias voltages used within the circuits described above are generated on-chip from a DC current input. These reference generators are incorporated into pad cells.

5.1.1.4 Implementation

The Test ASIC was implemented as a full-custom design and fabricated in 0.25-μm CMOS. Due to the nature of the goals for this chip it was pad limited as can be seen in the photograph in Figure 18 and thus had low circuit density. To gain more test points and allow for more test structures, an array of probe pads (seen at the center of the layout) was included in addition to the perimeter wirebond pads. Table 10 lists the tests that were planned for in the implementation of the Test ASIC. Different combinations of I/O pads and transmitter and receiver circuits were used to isolate the circuits under test.

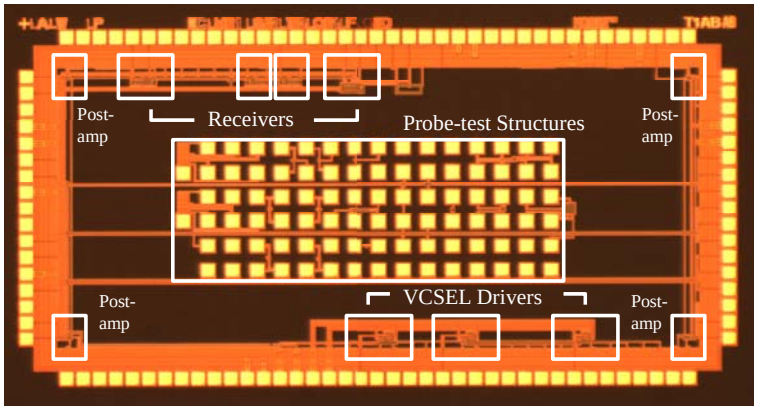

Figure 18. Annotated photograph of the Test ASIC die.

Multiple instances of the transmitter circuit include versions with storage within the cell for the bias DAC and modulation DAC settings. This is in the form of an eight-bit flip-flop based shift register. The transmitter circuit (pictured in Figure 19) is approximately 90 μm x 55 μm or 125 μm x 75 μm including the register file. Apart from the digital-to-analog converters and the differential pair, the transmitter cell consists of two resistor-capacitor filter circuits. These filters provide a clean local bias voltage to the cell. They are implemented using high-resistance polysilicon resistors and varactors that provide a high capacitance per unit area between polysilicon and the substrate of the chip.

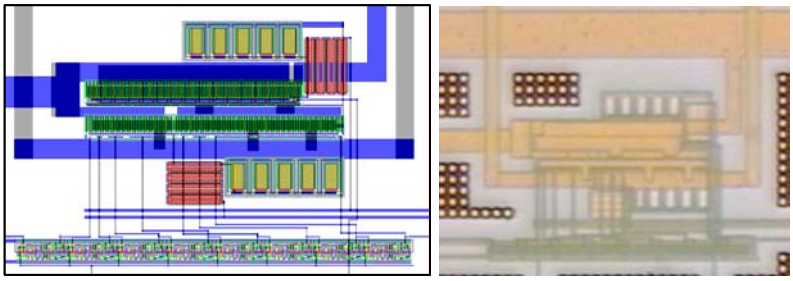

Figure 19. VCSEL drive cell. (left) Layout of driver including storage register at bottom. (right) Microphotograph of the same area on the fabricated die.

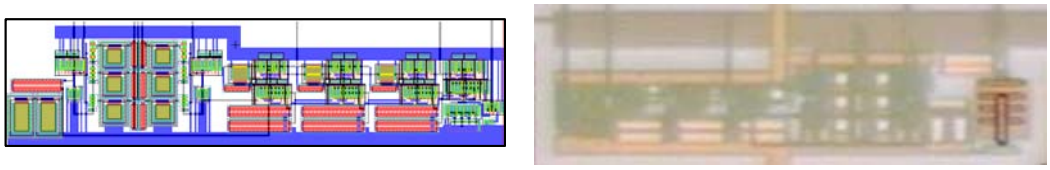

Figure 20. Receiver cell. (left) Layout. (right) Microphotograph of fabricated circuit.

Table 10. Test ASIC Verification Plan

Test #	Description / Path being tested
Probe pad tests	
P1	Multi-finger transistor characterization
P2	Resistor measurement
P3	Bias cell characterization
P4	Scan chain test
P5	ESD cell characterization
P6	Post-amplifier characterization
P8	LVDS I/O cell breakout
Perimeter pad tests	
1	Analog with ESD pad to Bare pad
2	Bare pad to CMOS out pad
3	CMOS in pad to CMOS out pad
4	CMOS in pad to LVDS out pad
5	LVDS in pad to CMOS out pad
6	LVDS in pad to LVDS out pad
7	CMOS in pad to Transmitter to Bare pad
8	LVDS in pad to Transmitter to Bare pad
9	CMOS in pad to Transmitter with shift register to Bare pad
10	Bare pad in to RX to CMOS out (with noise ring option)
11	Bare pad in to RX emulator to CMOS out pad
12	Analog with ESD pad in to RX emulator to CMOS out pad
13	Bare in to RX emulator to LVDS out

Four instances of the receiver cell are included on the Test ASIC with connections indicated in Table 10. Three of the four copies include a cell at the receiver front end to emulate its being connected to a photodetector. One of these is pictured in Figure 20. This layout is approximately 140 μm x 30 μm. The purpose of the emulator cell is to facilitate testing by allowing a voltage signal to be supplied to the chip rather than a current signal. An inverter based ring oscillator circuit surrounds the fourth receiver instance in order to allow for the impact of nearby switching digital logic on the operation of the receiver circuit to be tested.

5.1.1.5 Test Results

After fabrication, the Test ASIC was characterized using three different test setups. These were probe testing of the bare die, use of a general-purpose test fixture with packaged die, and use of a specialized chip-on-board test fixture. These three setups were used to carry out the planned testing. A series of test result data will be presented next.

Individually manipulated needle probes were used with a probe station to do the first tests using bare die. High-resistance polysilicon resistors (formed by silicide-blocked polysilicon) were used in a number of the custom cells and therefore were independently tested using probe contacts. The resistance measured for a 100 Ω resistor, a 1 kΩ resistor, and a 20 kΩ resistor were found to be within ±10% with the exception of one die which had a 100 Ω resistor that measured 81.4 Ω.

Other sub-circuits that were tested with the probe station included the bias-voltage generation cells for the LVDS outputs, receiver, and transmitter, as well as the post-amplifier cell of the receiver circuit. Results of these tests were plotted in Figure 21-Figure 23. The receiver bias generator included a diode-connected PMOS that sourced current back to the reference current supply (hence the absolute value bars in the plot) and the LVDS and transmitter bias generators were similar except NMOS based. All three bias generators were sized to operate with a nominal current of |1 mA| and therefore have a wide region of operation about that point wherein the bias voltage varies approximately linearly with current.

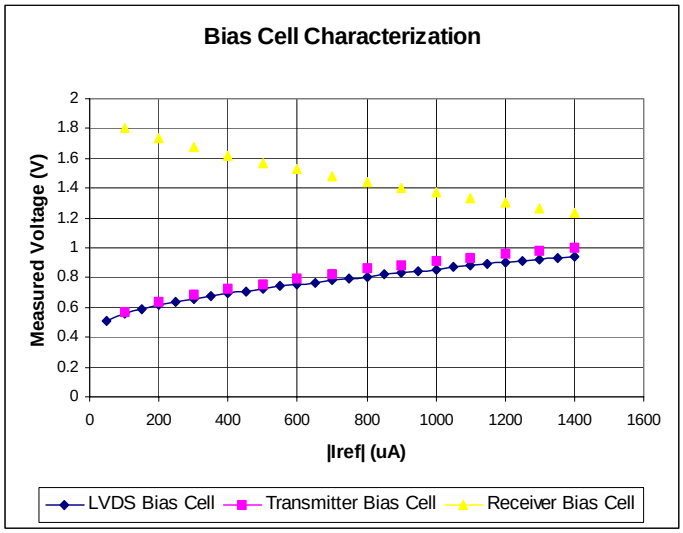

Figure 21. Test data from probe testing the bias voltage generation cells used in the LVDS output driver, the transmitter, and the receiver.

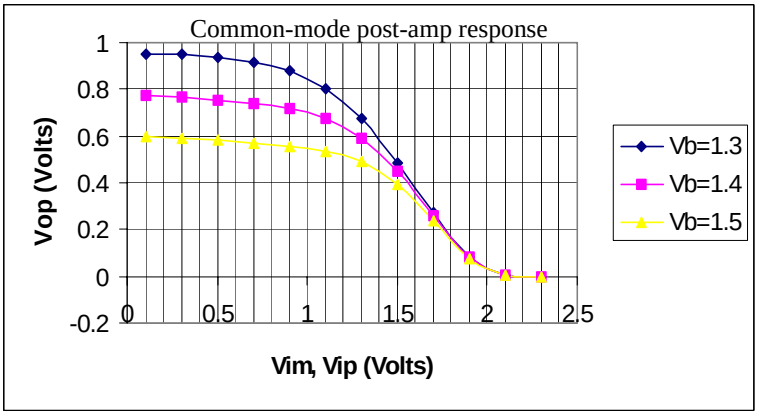

Figure 22. Common-mode test of post-amplifier stage of the receiver circuit. Inputs are tied together and swept for different values of the bias voltage vb.

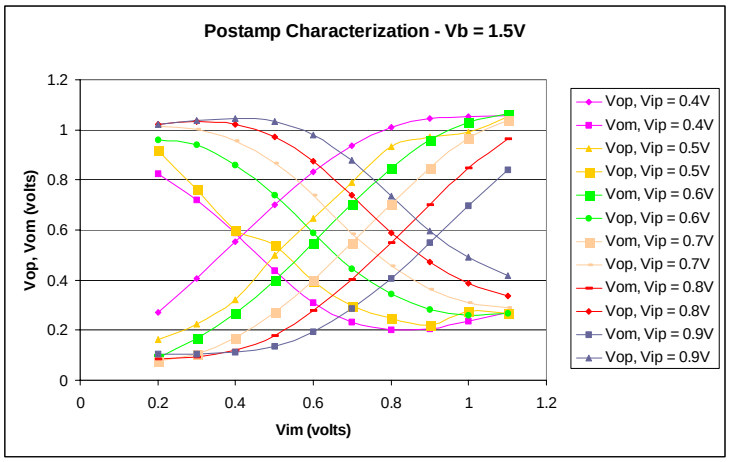

Figure 23. Differential-mode test of post-amplifier stage of the receiver circuit. Inputs are driven separately. As expected, vom and vop curves cross where vim equals vip.

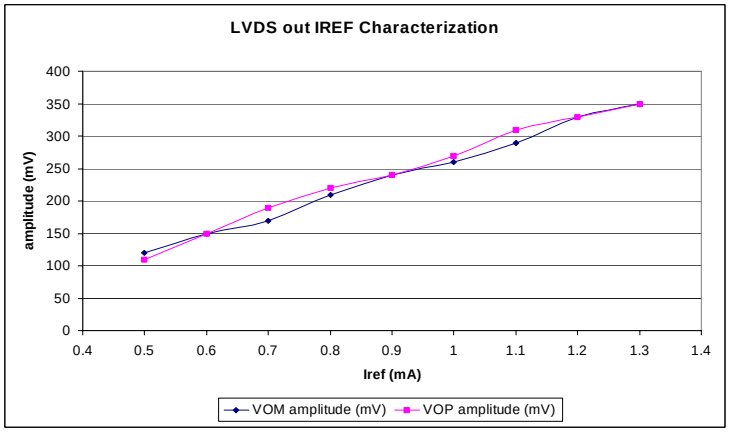

Figure 24. Test data from electrical characterization of the Test ASIC LVDS Output driver. The Iref input being swept controls the bias voltage of the CML buffer stages and LVDS output stage of the LVDS driver.

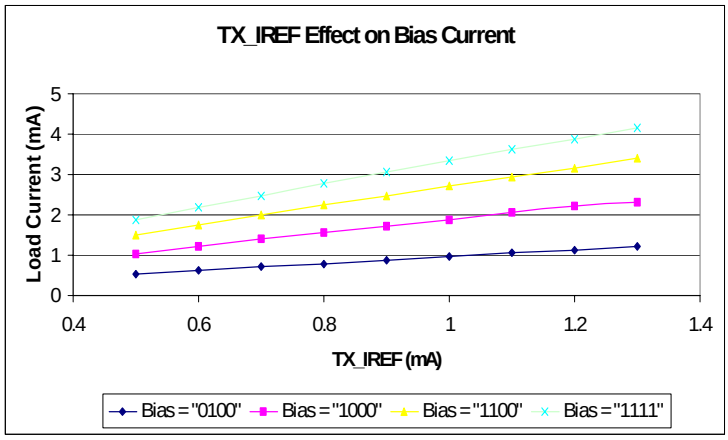

Figure 25. **Test data from electrical characterization of the Test ASIC VCSEL driver. The bias voltage is swept by changing the input bias current for multiple settings of the bias DAC. Note that zero bias is not plotted here.**

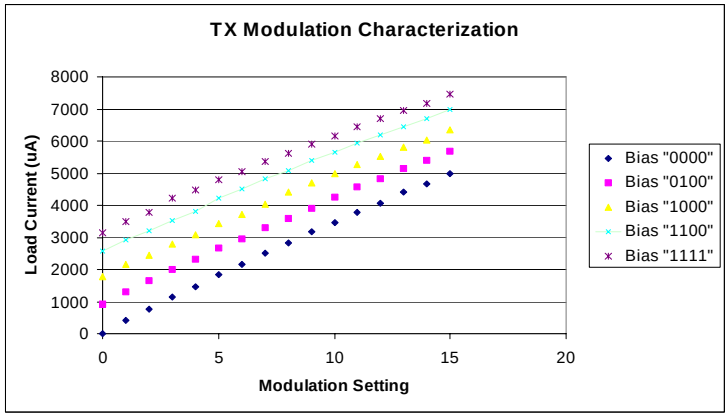

Figure 26. **Test data from electrical characterization of the Test ASIC VCSEL driver. The modulation DAC control is swept through its 16 possible settings for different settings of the bias DAC.**

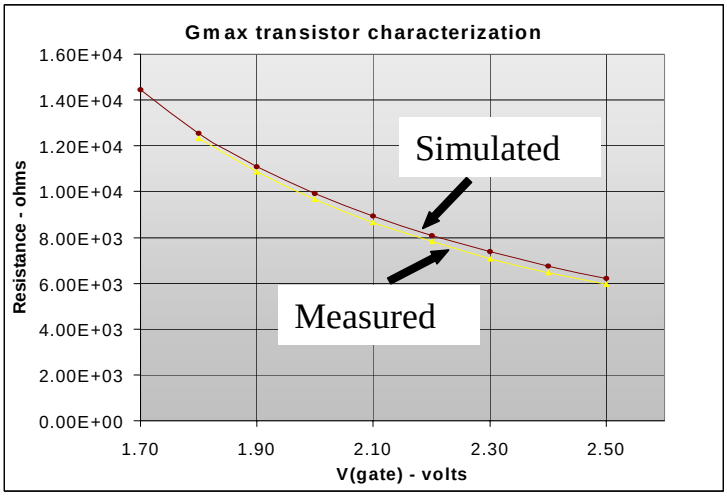

Figure 27. Test and simulation data showing DC characteristic of one of the feedback transistors within the receiver pre-amplifier. Good correlation is observed.

Five of the Test ASIC dies were packaged in pin-grid-array package in order to simplify testing of the circuits that were connected to wirebond pads at the chip periphery. A "universal test board" was then used to conduct the I/O and analog cell tests listed in Table 10 at low speed. All of the pads were found to function as desired with this setup and the transmitter and receiver cells were tested and characterized. Figure 24 illustrates the range of output voltage swing that was achieved in the LVDS output cell by sweeping the input bias current, which in turn controls the bias voltage for the LVDS driver. A 150-ohm load resistor was used to emulate a VCSEL device connected to the transmitter output in order to characterize the bias and modulation current as a function of the digital control settings. As shown in Figure 25 and Figure 26, the bias DAC was able to provide a continuous current in excess of 4 mA and the modulation DAC was able to provide up to 5 mA. This met

the current drive range specified at the time the VCSELs were being designed. The receiver circuits were first tested by comparing measured to simulated date for the pre-amplifier feedback and close correlation was observed as shown in Figure 27. Additional low speed AC testing of the receivers was done by using an arbitrary waveform generator to create a voltage signal that was connected to the receiver by way of the on-chip "emulator" circuit described earlier.

In order to perform at-speed testing of the LVDS input and output drivers a new test setup was required to ensure that the test environment did not corrupt the results. Therefore, a chip-on-board (COB) test vehicle was designed onto which bare die were mounted and wirebonded. This test setup is shown in Figure 28. In an effort to make the board design re-usable, every pad location was routed to connectors at the edge of the board using 50-ohm traces with options for high-speed co-axial connectors, AC coupling capacitors, termination resistors and low-speed pin headers. Two power and one ground plane surround the die-attach site for wirebonding power and ground pads. By using an arbitrary waveform generator to produce an LVDS signal, the path through the Test ASIC consisting of an LVDS input pad pair, on-chip single-ended interconnect, and an LVDS output pad pair was tested. Signaling at well beyond the 300 Mbps target rate was demonstrated as indicated in the 650 MHz signal shown.

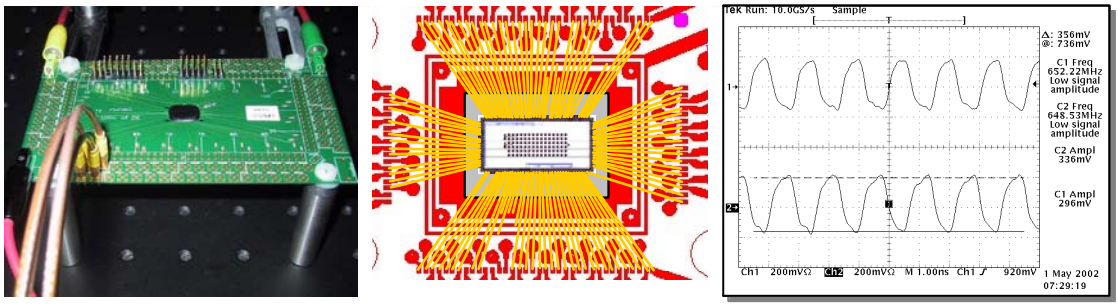

Figure 28. Chip-on-board test setup. (left) Photo of LVDS I/O test setup. (center) COB wirebonding. (right) Scope trace of LVDS 650 MHz output response.

5.1.1.6 Summary

The design and fabrication of an initial test chip was a valuable step in the progress toward the final VIVACE ASICs. The circuit topologies were verified and some points of improvement were identified and modified for the next chip designs. Additionally, as a result of experimenting with the Test ASIC an important decision in the input/output pad cells was made. Although all of the cells were found to be functional, their robustness to electrostatic discharge (ESD) events was less than desired for the full system. Evidence of ESD destruction was observed in the course of testing the chip. The pad cells developed for the Test ASIC did have a simple protection device, which experimentally demonstrated the ability to limit over-voltage events, but it was felt that the protection was not adequate for the fast transients and high currents of an ESD event. As a result, future VIVACE ASIC designs were based on a modified approach to the I/O pads that were based on the commercial cells but used customized input and output circuits

5.1.2 Transceiver ASIC

Plans for the Transceiver ASIC were made in order to have a backup for the Switch ASIC. This later migrated into a first-pass system build with a Switch ASIC-based MCM to follow and then to a replacement for the Switch ASIC in the final system. The primary goal for the Transceiver ASIC was the demonstration of the optical links that would be used by the free-space switch with limited digital logic and associated risk of protocol problems or complexity that might impact such a demonstration. Accordingly, the Transceiver ASIC was deigned to interface with the optoelectronic device array that was designed for the Switch ASIC.

The components of the Transceiver ASIC are largely those that were first tested in the Test ASIC with some modifications. A minimum amount of digital logic was added to allow all of the optical links within the demonstration system to be verified. The chip itself was designed and fabricated using 0.25-μm CMOS technology from TSMC. This is the same technology as described in Section 5.1.1. It is footprint compatible with the planned layout of the Switch ASIC, and thus with the bond sites of the MCM. It was submitted for fabrication in July, 2003 and testing of the fabricated bare die began in early October 2003.

5.1.2.1 Functionality

The primary functions of the Transceiver ASIC are to receive data from electrical input pads, use this data to drive the optical outputs, receive data from the optical inputs, and use this data to drive electrical output pads. To implement this transceiver functionality, custom circuits as well as standard digital library components were combined to form the architecture of the Transceiver ASIC.

5.1.2.2 Cells

Since the custom circuits used for the Transceiver ASIC are based on those fabricated in Test ASIC, only modifications will be discussed here. It is noted that although not discussed in the previous section, these cells would also have been used in the Switch ASIC implementation. The custom cells from the Test ASIC that were used are the VCSEL driver, photodetector receiver, LVDS driver, LVDS receiver, and associated bias-voltage generation cells. The custom pad library used on the Test ASIC was replaced with modified commercial library pads.

Test results of the VCSEL driver (transmitter) cell were favorable, showing the ability to provide adequate current to the VCSEL devices. The VCSEL specification called for the ability to provide a bias current of at least 1mA and a modulation current up to 6 mA. While it was believed that these were worst-case numbers, the over-drive capability observed in the Test ASIC was kept in order to have some operating margin. Therefore, the only modification to the transmitter cell was to incorporate the bias and modulation setting storage registers within it.

In the receiver cell, a few modifications were made in transitioning from the Test ASIC to the Transceiver ASIC. The first was an architectural change. The post-amplifier stage of the receiver has a differential input and thus requires a second input in addition to the one coming from the pre-amplifier stage that is connected to each photodetector. In order to generate this second input, another pre-amplifier, which serves as a reference-level generator is used. In order to ensure that this reference is as clean as possible and to guard across process variation across the die, the reference pre-amplifier was added to the base receiver cell rather than using a global reference pre-amplifier for the entire chip. Since the reference pre-amplifier is identical to the circuit connected to a particular photodetector, it is desirable to have a

DC current flowing into it that is mid-way between the photocurrent generated from a logic high and logic low. For a global reference pre-amplifier this is not a problem, but with the new dedicated reference pre-amplifier architecture and the large array size, the Transceiver ASIC required an additional on-chip current amplifier to drive the additional load. This circuit is a current mirror that delivers a current equal to one-fifth of an input DC current to every reference pre-amplifier.

In addition to including a second pre-amplifier with each receiver, the circuit itself was modified slightly. Feedback resistance in the form of NMOS transistors within the pre-amplifier stage controls the gain of the circuit. Adjustments were made to the sizes of these feedback transistors in order to tune the operating range based on expected photocurrent in the final system. The adjustments were made in such a way as to improve the expected uniformity across the receiver array.

For the LVDS input and output cells the main change was to remove them from the custom pad frame cells which were developed for the Test ASIC and include them within the core of the chip. The simple voltage clamp structures that were used as I/O protection devices in the Test ASIC were removed from the LVDS input cell. The on-chip 100-ohm termination resistor for the LVDS input was also modified in order to improve its accuracy.

Two issues that led to the development of the custom pad library for the Test ASIC remained to be solved in order to use the commercial pad library cells for the Transceiver ASIC. These were the availability of analog pads with ESD protection and the ability to bring high-currents through a pad-limited form-factor power pad. Analog pads were needed because of the analog current and voltage references used in the chip and also to allow the custom LVDS I/O cells to be used for

the high-speed inputs and outputs. Differential high-speed pad cells were not included in the library that was available for this chip design and the full-swing single-ended I/O that were available were determined to be unusable at the desired data rate. The high-current requirement for power and ground pads came from the number of optical channels per chip. It was determined that each power and ground pad needed to be able to reliably carry 80 mA in order to meet the worst-case power requirements of the internal cells and this was roughly double what the library cells could provide. In order to solve these two issues, the library pads were modified to create pads that could be used as an analog input/output pad without sacrificing the ESD protection structures already in them. The power pads were modified keeping the same narrow form-factor, but with added conductor width for greater current carrying capacity. The narrow form-factor was important in order for the size of the CMOS die to be kept close to the size of the optoelectronic device array.

5.1.2.3 Architecture

The architecture of the Transceiver ASIC was impacted to a large extent by the switch development. The die size, number of pads, and function of the pads were all fixed during the design of the Switch ASIC and before the design of the Transceiver ASIC began. This was done so that the MCM design could be done in parallel with the chip design. As a result, the Transceiver ASIC architecture is based on a 32-bit electrical input port and a 32-bit electrical output port. The optical ports are also the same as in the Switch ASIC, dictated by the optics design.

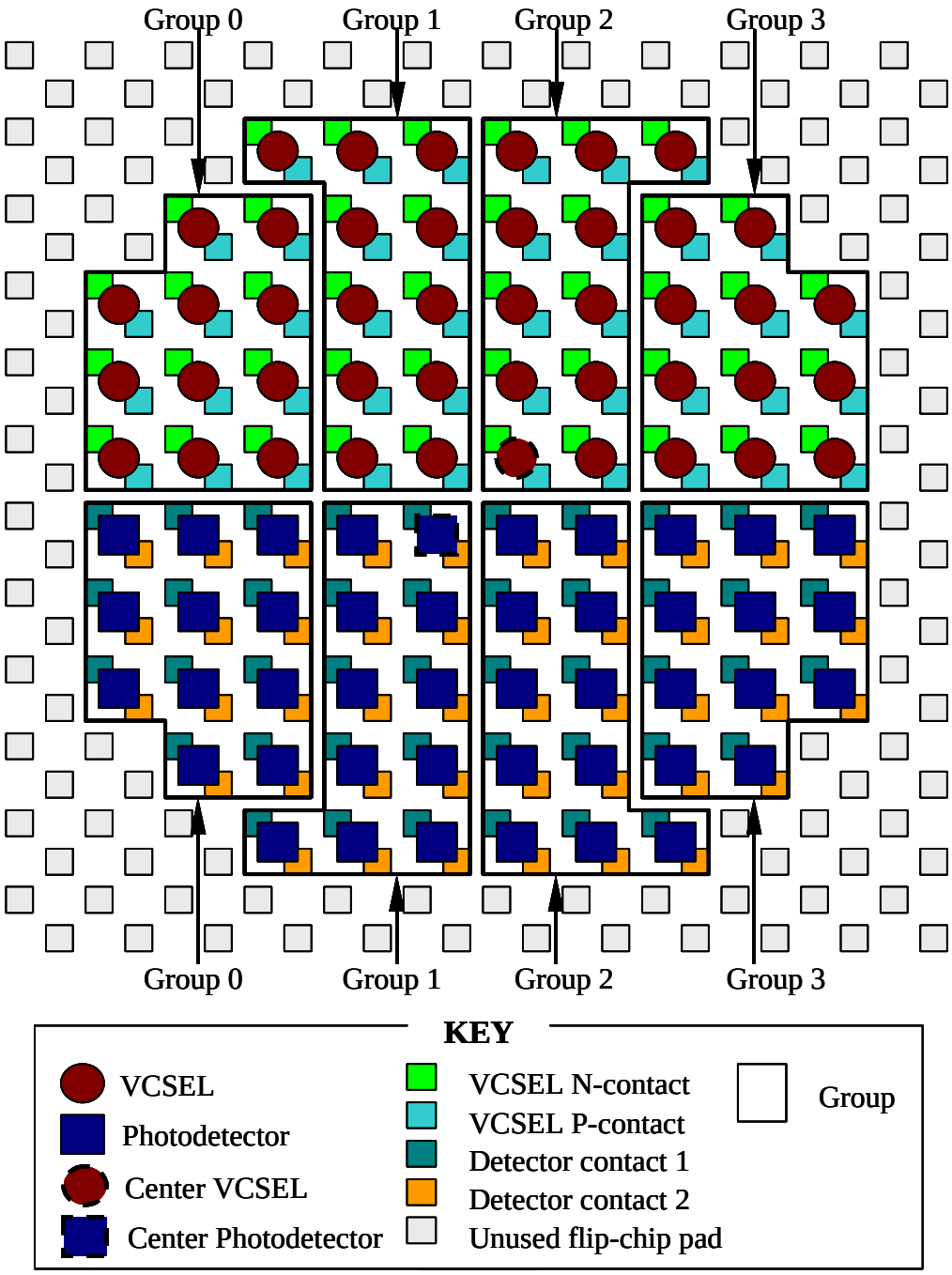

Figure 29. View of a single cluster showing optoelectronic devices and flip-chip pads.

The optical devices, and thus the transmitters and receivers, are grouped into clusters. Each cluster is a subset of a 12 x 12 group of optoelectronic devices, which is half VCSELs and half detectors. The cluster is made up of 44 VCSELs and 44 detectors. The three devices in the corners of a 10 x 10 array are not used, giving the cluster an octagonal shape, which most closely matches the circular shape of the optical lenses. One cluster constitutes an optical port on the Transceiver ASIC. Each cluster is further subdivided into four groups of outputs and four groups of inputs with eleven devices in each group. A complete cluster including flip-chip pads and optoelectronic devices is illustrated in Figure 29.

In order to demonstrate as many optical links as possible with minimal digital logic, the electrical inputs are fanned out to drive multiple transmitter circuits. Each of the 32 electrical inputs is used as the input to eleven transmitters. These input signals are buffered on the chip in order to drive the multiple transmitters and interconnect lines. On the receiver side, multiplexing is required to select which of the 352 optical inputs to send to the 32 electrical outputs. Again, each electrical output serves the eleven receivers in a group. The task of routing all of these lines was simplified by grouping the devices that share electrical inputs and outputs by their physical location on the chip. The architecture of a single cluster shown in Figure 30 is repeated for each of eight clusters on the Transceiver ASIC. These clusters are arranged in a 3 x 3 array, with one location unused.

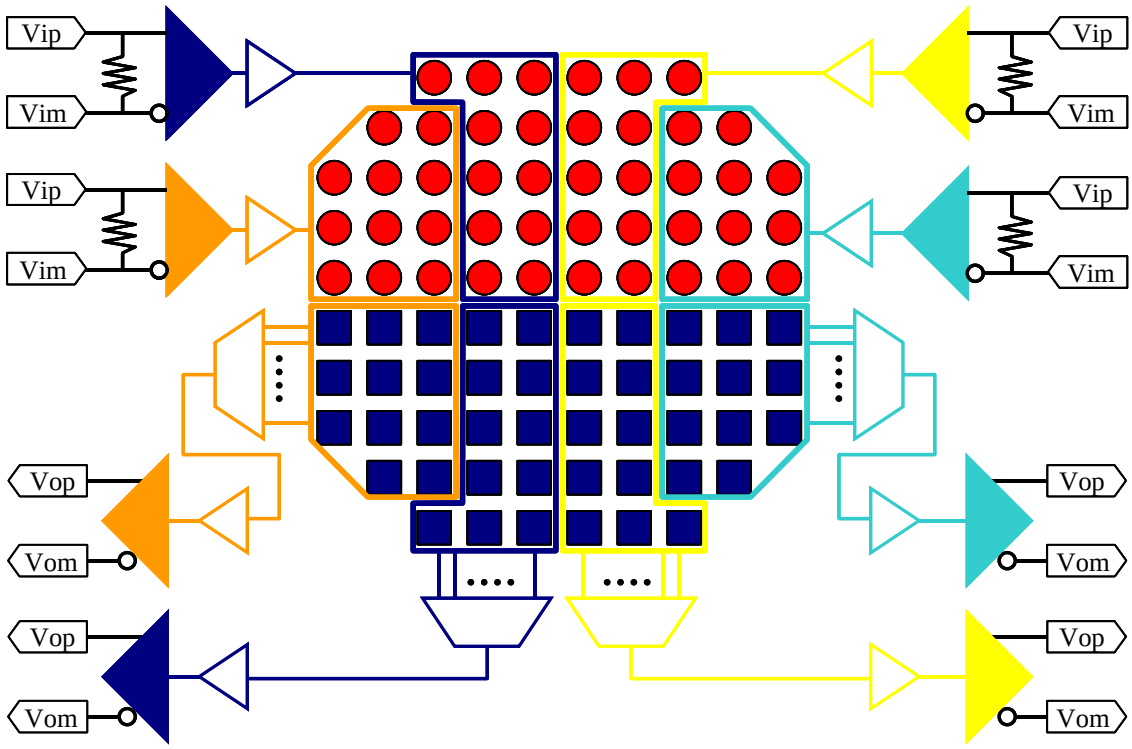

Figure 30. Cluster architecture for Transceiver ASIC. Each LVDS input pair connects to one digital buffer that drives 11 transmitter cells. The outputs of 11 receivers are multiplexed into a single signal that is buffered before driving a single LVDS output pair.

5.1.2.4 Interface

The electrical data interface to the Transceiver ASIC consists of the LVDS input and output lines described above, and two scan chain ports. There is one scan chain for configuring the bias and modulation current settings of the VCSEL drivers and one for the receiver enable controls and output multiplexor configuration. Each scan chain has an independent clock and reset signal in addition to the data input and output. These signals use CMOS-level single-ended I/O pads.

The power interface to the chip consists of a 2.5-volt supply, a 3.3-volt supply, and ground. There are also a number of bias voltages and currents. On the chip, all like voltages are connected together within the pad ring. This simplifies the ESD protection implementation. In order to provide isolation between the limited digital logic on the chip and the analog cells operating at the same voltage, independent routing from the power pads to the core logic is provided. Table 11 summarizes the power pad connections.

Table 11. Power Pad to Internal Circuit Allocation

Pad Voltage	Pad Names	MCM Power Plane	Circuit Connection
0 V	GNDDGc	GND	Core digital logic
	GNDDGo	GND	CMOS level I/O pads
	GNDLVDS	GND	LVDS I/O pads
	VSSA_TX	GND	VCSEL drivers
	VSSA_RX	GND	PD receivers
2.5 V	VDDc	VDD25	Core digital logic
	VDDLVDS	VDD25	LVDS I/O pads
	VDDA_RX	VDD25	PD receivers
3.3 V	VDDo	VDD33	CMOS level I/O pads
	VDDA_TX	VDD33	VCSEL drivers

5.1.2.5 Implementation

The Transceiver ASIC was implemented as a full-custom, 0.25-µm CMOS design containing approximately 270,000 transistors. Figure 31 is a photograph of the fabricated die. The active area of the design is 7.825 mm x 7.825 mm. The layout was created by placing the cells and routing them by hand using the layout editor, Ledit, from Tanner Research Inc. The scan chain for the transmitters was created by connecting the registers of the individual VCSEL drivers in series to form a 2,816-bit

shift register. For the receivers, which do not have a storage register within the cell for the enable signal, shift registers were created for each group. These registers consist of eleven bits for enable signals and four bits to hold the output multiplexor configuration bits. These receive-side configuration registers were connected in series to form one 480-bit shift register.

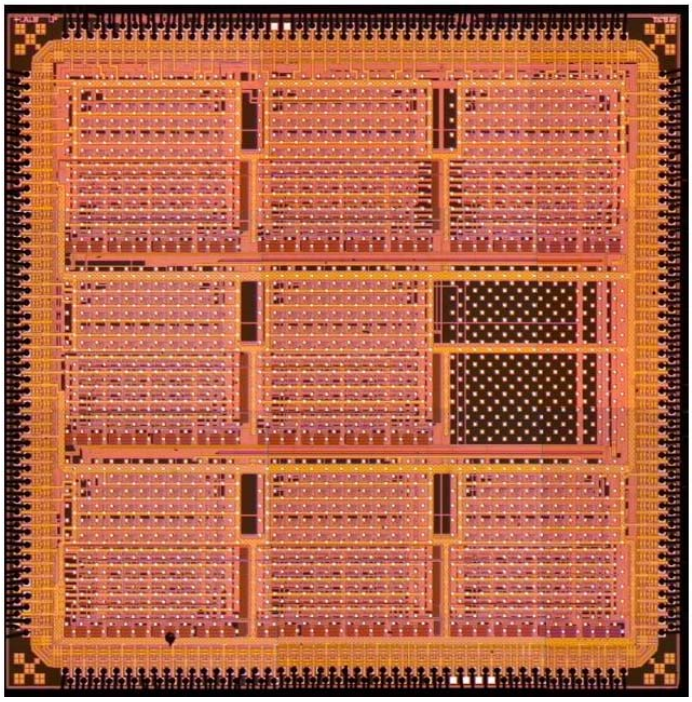

Figure 31. Composite microphotograph of 7.825 x 7.825 mm VIVACE Transceiver ASIC die (shown wirebonded to mechanical MCM without optoelectronic device array).

The pad frame is constructed from the modified commercial I/O cell library. It consists of 356 pad cells with staggered bond pads in two rows. The staggered bond pad layout was used in order to increase the effective wirebond pitch

to 150 µm. (Pad cells are placed with a 75-µm pitch.) The outer row of pads includes only power and ground pads and the inner row includes signal, bias and (limited) power pads. The outer-row pads are bonded to concentric rings for ground, 2.5 volts, and 3.3 volts and the inner-row pads are bonded over these to the ends of the signal traces on the MCM.

Top-level metal flip-chip pads are arrayed across much of the surface of the core of the Transceiver ASIC in order to attach the GaAs optoelectronic device array. These pads, which are smaller than the wirebond pads, provide the two contacts per VCSEL and photodetector required to operate the devices. The VCSEL array has isolated contacts for each device and the anodes are connected to a common power supply line routed on the Transceiver ASIC. Similarly, the photodetectors are isolated on the GaAs die and one contact of each device is connected to a common bias voltage line on the CMOS die. This bias line serves to ensure that the photodetectors are kept in reverse bias.

Table 12 shows the complete wirebond pad list for the Transceiver ASIC. Not all of the 356 pads that were allocated from the Switch ASIC are used here and so there are a few pads un-bonded on the MCM. The number of power and ground pads was calculated from a worst-case estimate of power consumption in the Switch ASIC assuming a current capacity of 80 mA for each pad. While this leads to a worst-case power consumption of nearly 12 watts for the VCSEL drivers alone, operating the chip under these conditions was not expected. Consideration of the expected VCSEL efficiency, power loss through the optical path, detector efficiency, and the receiver sensitivity indicated a need to use less than half this amount of VCSEL driver power in order to establish reliable links.

Table 12. Transceiver ASIC Pad List

Number of Pins	Use	Ring#	I/O Tech.	Function	Signal name
64	In	1	LVDS	32-bit Data input	datain_p/m
64	Out	1	LVDS	32-bit Data output	dataout_p/m
6	In	1	LVCMOS	Scan Chain	ScanIn, ScanClk, ScanReset
2	Out	1	LVCMOS	Scan Chain	ScanOut
2	In	1	LVDS	Fast Clock Input	fclk p/m
1	Out	1	Analog	Thermal sensor current output	Itherm
2	Out	1	LVDS	Clock Out	clkOut p/m
2	In	1	Analog	Corner VCSEL Control	cornerCtrl p/m
1	In	1	LVCMOS	Center VCSEL Control	centerCtrl
9	Out	1	Analog	Center Detector Monitor	centerMonitor
14	Pwr	0		Power supply for receivers	VDDA_RX
14	Gnd	0		Ground for receivers	VSSA_RX
6	V-DC	1	Analog	Analog gain control for receivers	RX_gain
2	V-DC	1	Analog	Detector Common Contact	Vdet_Common
1	I-DC	1	Analog	Receiver Bias	RX_Ivb
2	I-DC	1	Analog	Receiver Bias	RX_Ibal
47	Pwr	0		VCSEL Common / TX power	VDDA_TX
47	Gnd	0		Transmitter Ground	VSSA_TX
1	V-DC	1	Analog	Transmitter reference voltage	TXRef
1	I-DC	1	Analog	Transmitter reference current	TX_Iref
4	Pwr	0		VDD for LVDS cells	VDDLVDS
4	Gnd	0		VSS for LVDS cells	GNDLVDS
1	I-DC	1	Analog	LVDS Output reference voltage	LVDSREF
1	In	1	LVCMOS	LVDS Output Enable	LVDSENA
20	Pwr	0		Digital Power for Core	VDDc
4	Pwr	0		Digital Power for I/O	VDDo
20	Gnd	0		Digital Ground for Core	GNDDGc
4	Gnd	0		Digital Ground for I/O	GNDDGo
2	I/O	1	LVCMOS	CMOS-level feedthru test	tdi, tdo
6	--	1		Reserved	RSV(5:0)

5.1.2.6 Test Results

After the fabrication of the Transceiver ASIC, the initial testing of the bare silicon die was done prior to sending dies to be hybridized. For this testing, a probe station with individually positioned probe needles was used. Several of the same tests and characterization steps that were done in the testing of the Test ASIC were repeated with this setup. The input/output pads which were customized from standard library parts and the high-speed electrical input/output cells where tested and verified as was the digital logic which sets up control lines for the independent VCSEL drivers and photodetector receivers. This testing was successful with all circuits responding as expected.

More extensive testing of the non-hybridized die was done using the first mechanical version of the MCM as a package. Wirebonding onto this MCM allowed for access to more pads at once, as required to test the analog transceiver cells. By powering the chip through solder connections to the MCM traces more probe connections could be made. One of the primary testing goals, which was achieved using this setup, was the test and characterization of the 352 VCSEL drivers on the Transceiver ASIC. The flip-chip pads that connect to the transmitter outputs were probed one by one. A precision current meter was used to measure the current through a 150-ohm resistor that was connected between the probe needle and a 3.3-volt DC supply. This resistor was used to emulate a VCSEL device being connected in the same configuration, as it would be in the hybridized chip. Figure 32 shows the result of this test for each VCSEL driver set to its maximum bias current setting and zero modulation current setting. The modulation DAC was not characterized, and therefore disabled, due to a limitation in the number of probes that could be used at one time. (Modulation current is dependent on the individual data inputs, which were

not simultaneously tested with this setup.) The data in this graph is plotted against the physical location of the driver on the die. Therefore, the VCSELs within the cluster are visible and the spaces where unused devices (i.e. between clusters and in the center-right cluster which is not used) or receivers are located are represented by the floor of the surface. As shown, all VCSEL drivers were found to be functional with good uniformity across the die. One thing to note is that there is some unknown but small variability introduced by the contact resistance when each driver output was probed.

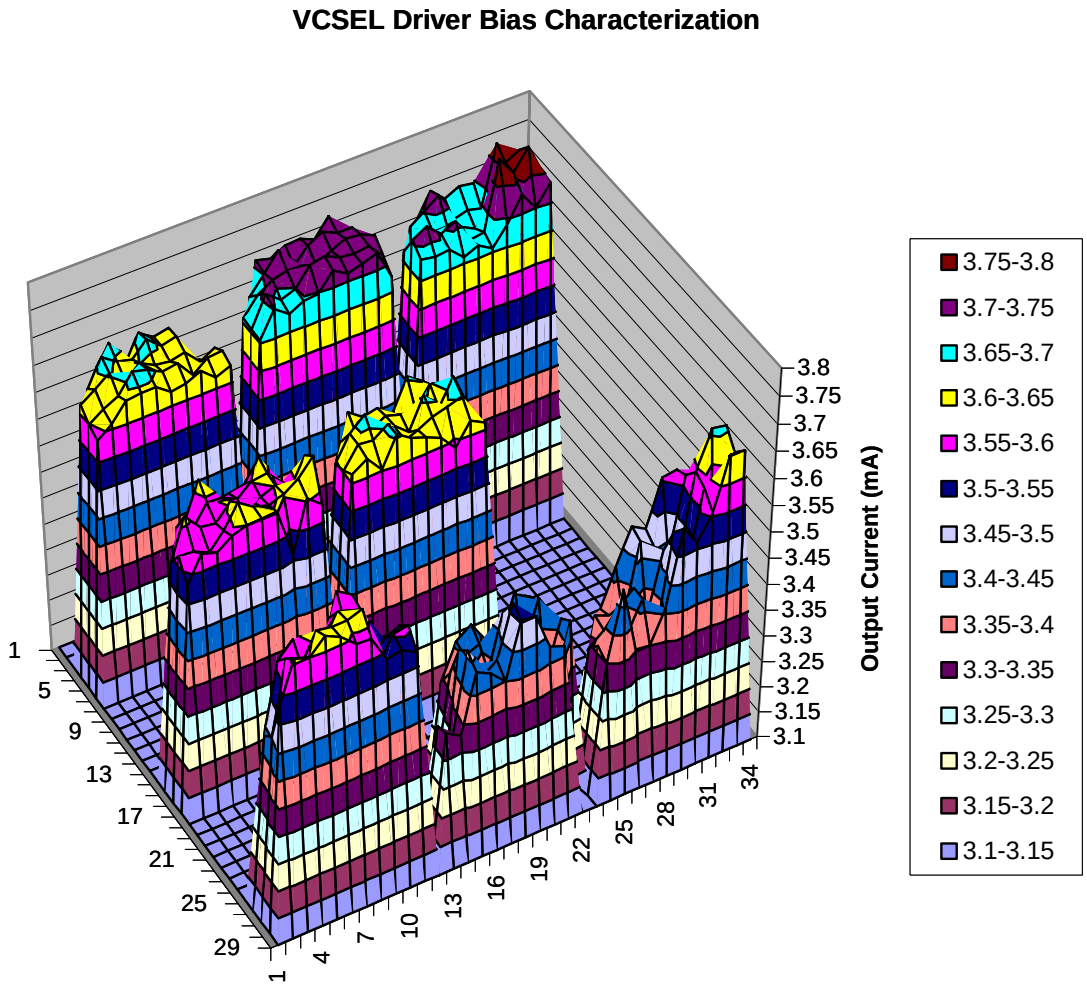

Figure 32. Plot of VCSEL driver uniformity test data. For maximum bias current setting, less than a 10% variation in output current observed across the entire die.

5.1.2.7 Summary

The Transceiver ASIC, which was designed as a backup to the Switch ASIC or first-pass system IC, became the centerpiece of the VIVACE hardware demonstration. It successfully provided the ability to drive all 352 VCSELs and

monitor the outputs of all 352 photodetectors that made up the active area of a single VIVACE optoelectronic device array. The design of the Transceiver ASIC was thus a subset of the Switch ASIC design only in terms of the digital logic implemented and otherwise provided full compatibility with the optoelectronic devices, the MCM, and the Motherboard.

5.2 System Description

The electronic hardware of the VIVACE experimental test bed and final demonstration system consists of an eight-site multi-chip module mounted onto a printed circuit board, which serves as the system Motherboard. The optical and opto-mechanical system is supported by the mechanical assembly that secures the MCM to the Motherboard. The MCM was populated with eight hybridized Transceiver ASICs. A ninth site visible on the MCM in Figure 33 was not actually implemented due to the difficulty in routing and therefore is not populated. This simplification was also made in the layout of the Transceiver ASIC, but unfortunately an incompatibility between the specific site not populated on the ASIC and MCM effectively reduced the number of chips in the final system to seven. As shown in Table 13, this brings the total number of complete links possible in the system to over two thousand. The final system only uses seven macro-lenses corresponding to the seven SPAs that can be fully linked, but a number of links that have only a VCSEL or a detector exist as indicated in the table.

Table 13. Full-System Optical Link Summary

Specification	Total	Effective
Links per Cluster	44	44
Clusters per SPA	8	7
SPAs per MCM	8	7
Links per SPA	44*8 = 352	44*7 = 308
Links per MCM	352*8 = 2,816	308*7 = 2,156
Single-ended Links per MCM	44*8*7 = 2,464	N/A

5.2.1 MCM

The multi-chip module for the VIVACE system was designed and populated by a partner research group at the Mayo Foundation. It is a 5 inch x 5 inch low-temperature co-fired ceramic (LTCC) substrate with die-bond sites for eight hybridized SPAs. The large overall size coupled with 2-mil trace / 3-mil space traces is very aggressive for this technology and led to fabrication delays. LTCC is often used for high-frequency applications, including microwave and wireless, based on its low dielectric constant and loss. Its multi-layer capability and good surface flatness also make it well suited to use in an FSOI module. A customized, high-precision placement process achieved with a commercial flip-chip bonder was used in the assembly of the MCM. The resulting placement accuracy of less than ± 5 μm (across the 5-inch surface) was more than adequate to meet the requirements of the optical alignment.

The interface between the MCM and each CMOS ASIC consisted of three pseudo power planes with local decoupling capacitors and a ring of 180 wirebond sites. These wirebond sites are, in turn, routed to an array of land-grid array pads at the edge of the MCM that is used in conjunction with a high-speed connector to mount

the MCM on the system Motherboard. The MCM is shown in Figure 33 and the connector between the MCM and Motherboard is illustrated in Figure 34.

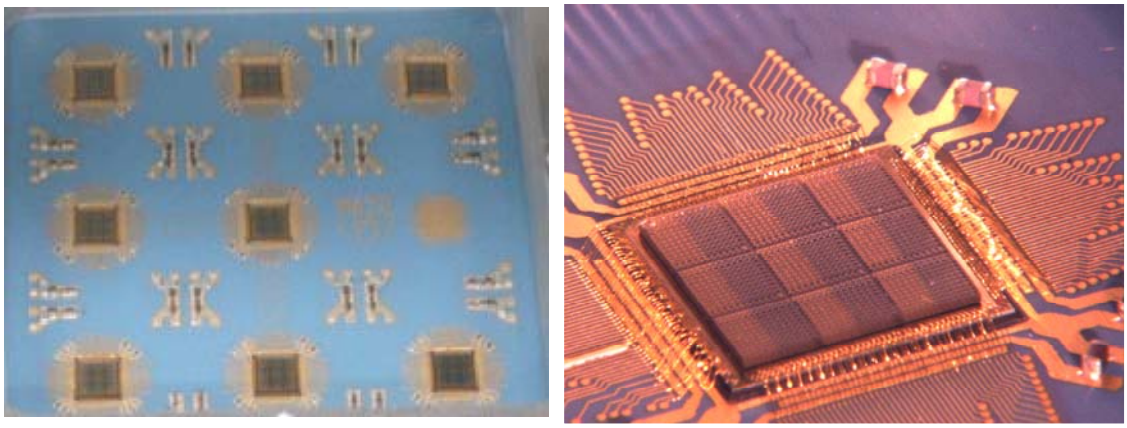

Figure 33. Fully assembled MCM and close-up of single SPA.

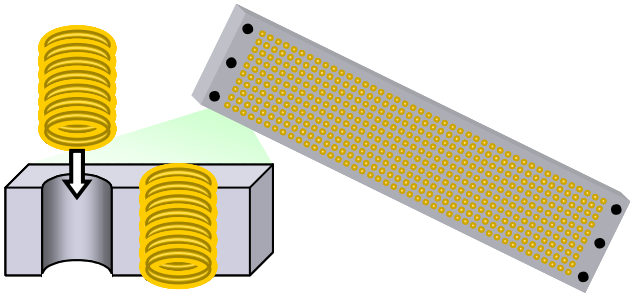

Figure 34. Interposer LGA connector. (left) Cut-away view of gold wire inserted into hole in insulating material. (right) View of full interposer with 44 x 10 array of conductors.

5.2.2 Motherboard

As both the electrical and mechanical base for the remainder of the VIVACE hardware, the Motherboard is the base piece of the VIVACE experimental test bed. The PCB itself is a twelve-layer copper on FR-4 substrate. It has been manufactured using copper features sizes of 5 mils with 5-mil spacing. In order to accommodate all of the components needed and for the ease of use and flexibility during the VIVACE experimentation, the Motherboard is quite large at 34.5 cm x 34.5 cm. While this makes the whole test bed seem large, it should not be assumed that this could not be made much smaller if implemented as a specialized commercial or military product.

5.2.2.1 Functionality

During the course of the VIVACE program, the final demonstration goals were modified such that the Motherboard-based test bed would be self contained rather than interfacing with a number of VONIC cards by way of parallel optical fiber-ribbon inputs and outputs. The original architecture thus called for parallel fiber transceiver modules and SERDES devices in a complimentary configuration to those on the network side of the VONIC. However, in the final Motherboard design these components were not required and instead all data generation and termination occurs on the Motherboard itself.

The use of the Transceiver ASIC as the silicon design included in the final system impacts the motherboard functionality requirements. Had the Switch ASIC been used for the final system, the Motherboard would have emulated a number of host computers performing a distributed computation and communicating via the free-space switch. Since this was not required, the Motherboard functionality was

simplified to the verification of optical links. The design of the Motherboard was carried out at the same time as the Switch ASIC and so there are a number of design and architecture decisions that were made based on the design of the switch. A fundamental test that is ubiquitous in the fiber-based optical communication market is Bit Error Ratio Testing (BERT). This BERT functionality is thus the base functionality for the VIVACE Motherboard. However, the uniquely massive parallelism of the optical links effected by the VIVACE hardware requires a parallel implementation which is not readily available with off-the-shelf test equipment. At the interface between the Motherboard and the MCM there are sixteen electrical ports, eight of which serve as inputs to the MCM and eight as outputs from the MCM. Each of these ports consists of thirty-two differential pairs. This means that on the Motherboard there is a need to simultaneously source and sink 256 channels of data in order to implement the desired parallel BERT. This forms the basic functionality that has been implemented for the final demonstration.

5.2.2.2 Architecture and Components

The architecture of the VIVACE Motherboard is dominated by five components: the MCM and four Field-Programmable Gate Arays (FPGAs), which are re-programmable hardware devices that can be configured into user-defined logic circuits. FPGAs are used in this design because they afford the flexibility to change the hardware configuration and logical operation quickly. This makes them ideal for use in a prototype evaluations system, which may require different functionality for different tests. The particular FPGA in this architecture is an XC1000E-6FG680 manufactured by Xilinx Inc. This device has an approximate capacity equivalent to one million logic gates, has 512 user configurable input/output pins, which can be

configured as up to 246 differential pairs, and is packaged in a fine-pitch ball grid array package. Each of these four FPGAs is used to interface with two of the eight ASICs on the MCM. Therefore, the FPGAs are each configured with sixty-four LVDS inputs and sixty-four LVDS outputs.

The MCM is, from the perspective of the PCB design, a 12.7 cm x 12.7 cm component with 3,520 electrical connections and a 16.5 x 16.5 cm keepout area. Needless to say, this physical size and number of connections created some implementation challenges in the design of the Motherboard. The MCM is attached to the Motherboard using dematable connectors, so that it could be assembled in stages, re-worked, or exchanged. The connection is made using eight Land Grid Array (LGA) "interposers" arranged in a square matching the outside edge of the MCM. A three-inch square cutout is in the center of this area allowing heat sink contact to the backside of the MCM. Each interposer contains 440 electrical conductors made from a coiled gold wire inserted into a hole through the insulator as shown in Figure 34. The pads on the PCB to interface with these conductors were each 22 mils in diameter and placed with a 40-mil pitch. Alignment pins are used to precisely position the MCM onto the interposers and Motherboard. The alignment holes on the Motherboard were specially processed as an added fabrication step with +/- 2-mil positional accuracy to ensure that the interposer conductors properly contacted the LGA pads on the PCB. A further restriction was placed on the PCB top-layer routing so that no traces were permitted within the 10 x 44 array of LGA pads that make up each interposer footprint. This change greatly reduced the risk of interposer wires shorting to nearby traces. In order to meet this restriction, silver-epoxy filled via-in-pad technology was used so that routing vias could be placed directly under the LGA

pads on the PCB. These restrictions proved to be very worthwhile, if not essential, in reliably assembling the system.

The majority of the architecture of the Motherboard can be divided into four pieces that are identical. Each of theses pieces contains one FPGA with associated components and interfaces to two ASICs on the MCM. This is illustrated in Figure 35. In addition to the MCM connections, each FPGA is connected to a number of pin-headers, jumper blocks, and switches. There are also two hexadecimal displays connected to each FPGA for the display of status/diagnostic information. Some lines connecting to the MCM can be controlled either by the FPGA or from pin-headers. These external connectors are for connecting a logic analyzer to the system. A single clock input is fanned out to each FPGA and Transceiver ASIC by way of an LVDS buffer on the Motherboard. Configuration data is required to program each FPGA and is stored in two non-volatile memory chips that are themselves programmed by a computer via a JTAG interface.

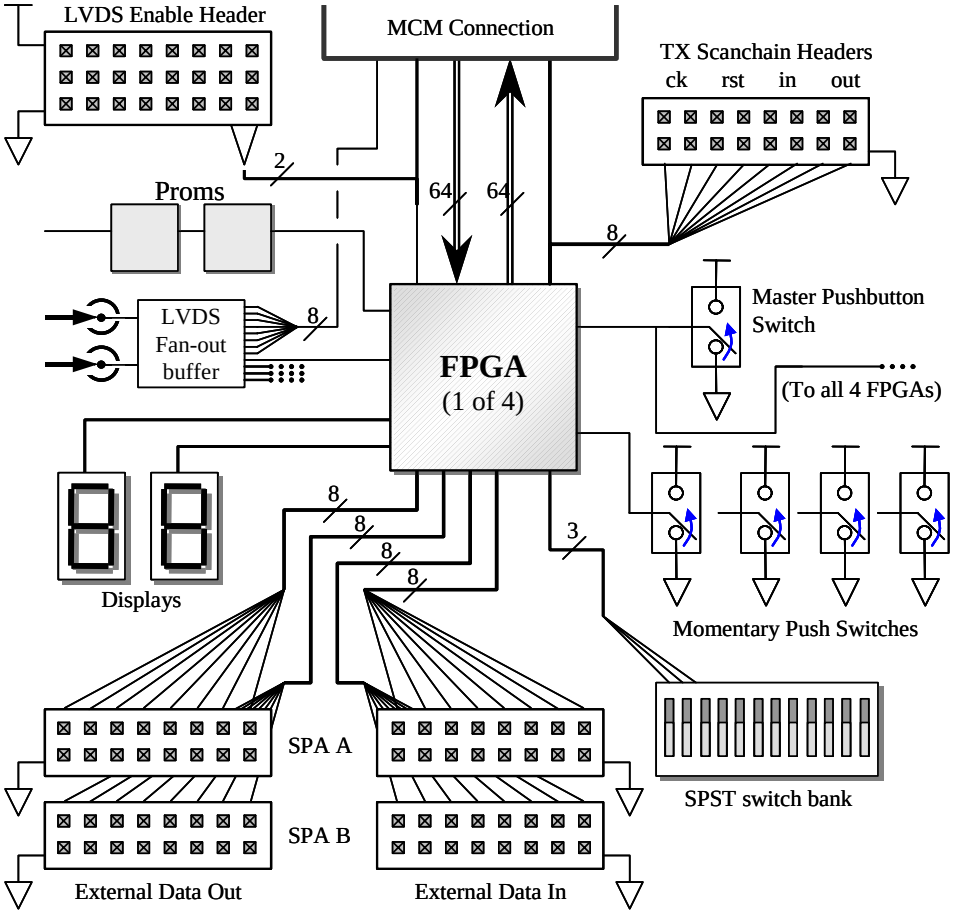

Figure 35. Architecture of Motherboard. One-quarter view showing connections to one FPGA.

5.2.2.3 Implementation

The Motherboard implementation process consisted of computer-aided design work, fabrication, assembly, and test in order to prepare it for integration with the MCM. Allegro software from Cadence Design Systems Inc. was used to generate the artwork files that were submitted for fabrication. The majority of the routing was done with an automated routing tool called Specctra. Special constraints were set up

in order route the LVDS signals as pairs with consistent trace-to-trace spacing and lengths in order to maintain 100-ohm differential impedance for these lines. Due to the number of traces to be routed and the highly congested areas surrounding the FPGAs and interposers, some manual routing was required. After fabrication of the PCB substrate by a commercial PCB vendor, a two-step assembly process was carried out to complete the Motherboard. The assembly of the FPGAs, which are BGA devices, was outsourced to a third party where a BGA re-work station with split vision system was used to place the parts and then the entire board was put through a solder re-flow oven. The remainder of the components (nearly 3000) was solder assembled by hand before electrical testing of the FPGAs. This testing was performed to ensure connectivity and programmability of the devices. The Motherboard substrate is shown in Figure 36.

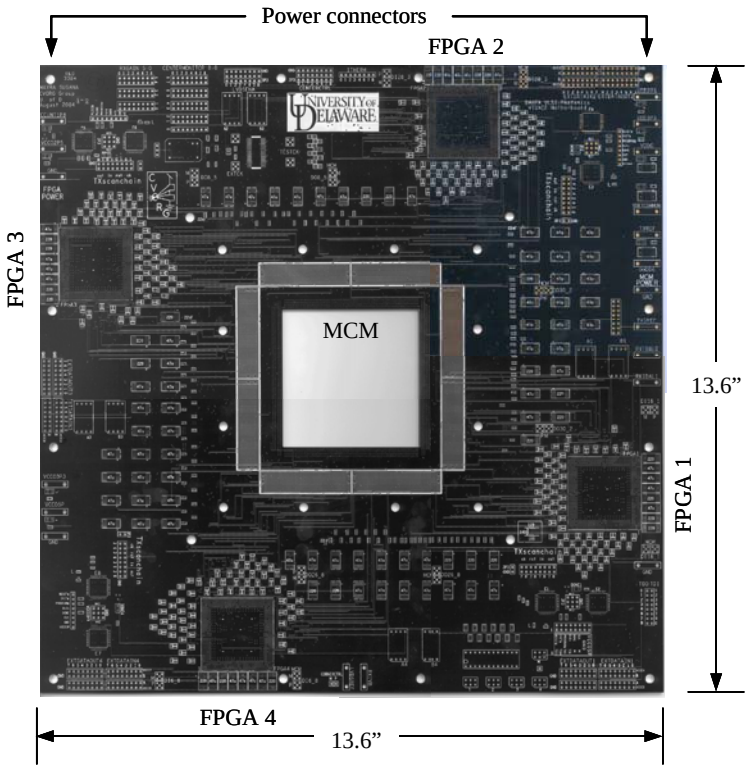

Figure 36. VIVACE Motherboard substrate showing sites for FPGA and MCM attachment.

5.2.3 Network Interface

A part of demonstrating the feasibility of the VIVACE system and a step in the full switch development is the custom network interface. This component is responsible for interfacing a host computer to the optical switched network and is called the VIVACE Optical Network Interface Card (VONIC). It is a copper and FR-4 PCB consisting of eight layers of routing and power planes. The host-side interface is a 64-bit / 66 MHz 3.3 volt PCI bus. The primary components on this board are a large FPGA, three serializer/deserializer ICs and twelve-channel fiber-optic transmitter and receiver modules.

Due to high power consumption, the VONIC is not powered from the PCI bus as most PCI cards are. Rather, an external connection to the host PC's power supply is made and the voltages needed for the VONIC parts (1.8, 2.5, and 3.3 volts) are generated on the card itself. The power consumption of this board results largely from the three AMCC chips used to generate serial data streams, which in combination draw almost 10 watts. Additionally, thermally enhanced chip packages, fin-type heat sinks, highly efficient DC/DC converter circuits, and forced-air cooling have been used to maintain reliable operation. A standard 4-pin power connector is used to connect the VONIC directly to a PC or workstation power supply. Isolation of the VONIC power from the PCI bus ensures a reliable means of providing power to the card without adversely affecting the ability to add other system components to the host's PCI bus. The assembled board is shown in Figure 37.

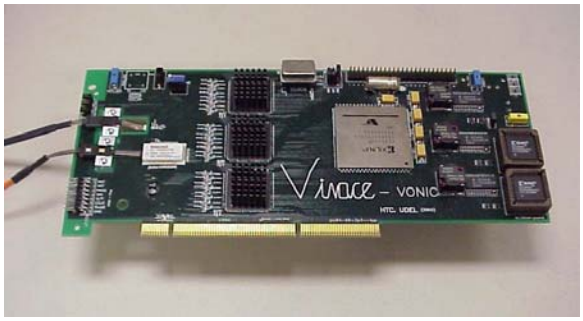

Figure 37. VIVACE Optical Network Interface Card (VONIC). Board dimensions are 25.7 cm x 10.7 cm x 157 mm.

The VONIC serves to provide the first protocol translation within the VIVACE network. It is capable of accepting and transmitting messages to and from

the host application as well as generating pseudo traffic for benchmarking purposes. This card has been built and demonstrated successfully [57][58][59]. The final system demonstration for VIVACE, however, incorporates the functionality needed to transmit data through the optically interconnected MCM, and therefore, does not use the VONIC hardware.

5.2.4 Optics

The custom optical system designed to implement the global free-space optical connection pattern was designed by collaborators at Applied Photonics Inc. and George Mason University (later at the University of Delaware). A novel, multi-scale lens system composed of custom optics and optomechanics was created to provide misalignment tolerance and no distortion. Distortion and misalignment are import in this type of system because of the relative size of the MCM (10 cm) and the photodetectors (60 μm) in order to enable the optical alignment process. This approach consisted of micro-lenses, mini-lenses, and macro-lenses named for their respective sizes. One micro-lens is used for each VCSEL and photodetector. The purpose of these lenses is to reduce the divergence angle of the VCSELs, effectively reducing their numerical aperture and thereby simplifying the macro-optics. It is desirable to place the micro-lenses very close to the apertures of the optoelectronic devices with good x-y placement accuracy. To achieve this, the micro-lenses are fabricated directly on the superstrate of the GaAs wafer by MicroFab Technologies Inc. using a process similar to ink-jet printing [60][61]. The next scale of optics is the mini-lens. This is an approximately 2 mm diameter aspheric lens placed over each cluster of optoelectronic devices. It serves to perform bean steering to once again simplify the macro-lens design and enhance the overall performance of the optics.

The final stage of optics, the macro lenses, implement a 4-f infinite conjugate optical system. They are composite four-element lenses on the scale of the hybrid chips on the MCM with one placed over each SPA. The composite macro-lens was packaged in a metal barrel with the mini-lenses attached directly to the front surface of the macro-lens. The macro-lenses are responsible for providing the global, all-to-all connectivity for the system. The interconnect pattern is folded back onto itself by way of a mirror placed at an appropriate distance (roughly 15 cm) from the top surface of the macro-lenses [62][63].

The lenses in the system are positioned and supported by a custom opto-mechanical system. A lens plate with slots for each macro-lens barrel is supported by a positioning plate. This plate allows fine vertical and leveling adjustability for the plane of mini- and macro-optics and can be locked in place after positioning. Rotational positioning of the mini- and macro-optics as a unit is also achieved with this plate. The rigid mechanical connection between the optomechanics and the MCM is created by attaching four vertical posts to the pressure plate that holds the MCM in place on the Motherboard. The lens-pisitioning plate attaches to these posts at the bottom in order to hold the lenses very close to the surface of the SPAs. A mirror holder at the top of these posts completes the opto-mechanical assembly and allows the mirror to be adjusted for tilt and vertical position.

5.2.5 System Assembly

The VIVACE test bed was assembled onto the completed Motherboard at Mayo as follows (depicted in Figure 38). A mechanical sub-assembly was made to support the Motherboard using an optical breadboard and steel posts. An aluminum plate which is 16.5 cm x 16.5 cm x 2.54 cm with a 8.9 cm square cut-out in the center

and alignment-pin holes matching the location of those in the PCB and interposer was placed behind the Motherboard with a Teflon washer to protect the surface of the PCB. Pins were inserted into these alignment holes and the interposers were placed over the pins onto the front side of the PCB. The MCM was next placed onto the interposers, again using the alignment pins to ensure placement accuracy. Finally another Teflon washer and matching aluminum plate were placed on top of the MCM. Twelve holes through the top aluminum plate and PCB with matching threaded holes in the bottom aluminum plate allowed even compression force to be applied across the eight interposers in order to make electrical contact on all 3,520 connectors. Great care was taken during this process to ensure that none of the gold wire from the interposers protruded so much as to contact an adjacent pad on the MCM or PCB. Although the process went fairly smoothly, it did take one or two iterations to complete.

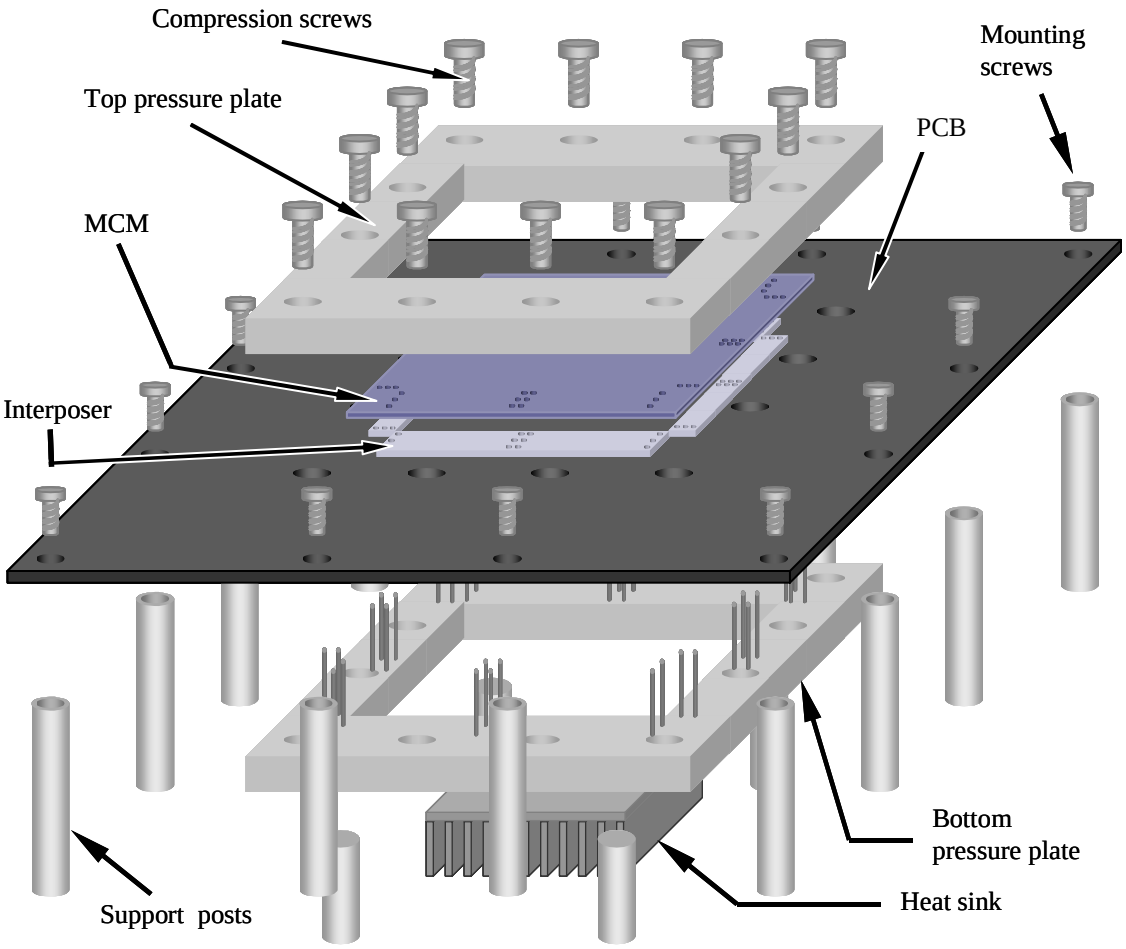

Figure 38. MCM to Motherboard assembly.

The electronic hardware in the VIVACE demonstration system was designed for use in a laboratory test environment. This allows flexibility during testing, reduces risk, and speeds the design cycle. As a result, several pieces of laboratory test equipment are used in the test-bed setup. Among these are: DC voltage supplies, DC current supplies, an arbitrary waveform generator, digital pattern generator and logic analyzer, and multi-channel oscilloscopes.

The DC voltage supplies separately provide power to the components on the Motherboard and the MCM as well as the required bias voltages for the Transceiver ASIC. Similarly, DC current supplies provide bias current to the Transceiver ASIC circuits. Some of these bias voltages and currents can be separated per-ASIC, but can also be driven from a global source in order to reduce the amount of laboratory equipment required. An arbitrary waveform generator (AWG) is used to generate an LVDS signal that serves as the master clock for the Motherboard and MCM. The Motherboard also has the option of on-board clock oscillators, but the use of the AWG simplifies the testing due to the ease of changing the frequency and characteristics of the clock signal. The pattern generator and logic analyzer are central to the test bed control because of the vast configurability of the Transceiver ASIC. The oscilloscope was primarily important during the optical alignment process where it was used to monitor analog outputs from the chips.

The system assembly was completed by aligning the optical elements with respect to the SPAs and each other. The procedure consisted of a series of course and fine adjustments carried out using visual inspection and electrical performance monitoring. Course adjustments included rotational positioning of the macro-lens barrels within the slots in the lens holder plate, rotational placement of the lens holder within its support plate, and positioning of the folding mirror. Precision translation stages were used to perform fine adjustments on the position and rotation of the lens holder plate as well as the lens barrels within the slots. Once aligned, the optics were mechanically locked in place.

Determination of the optical alignment was made in two ways. First, a reflective neutral density filter was used as a beam splitter in order to visually observe

that the VCSEL beams were landing on the micro-lenses of the appropriate detectors. Figure 39 shows a view of an aligned cluster. The faint reflections from the surfaces of the detector micro-lenses can be seen in the right half of the cluster. A view of what an entire SPA "sees" when all other SPAs transmit to it is also shown in this figure. Next, fine-tuning of the alignment was made by observing and optimizing the analog electrical output of selected receiver circuits. Alignment of a given lens was completed when simultaneous connections between a given SPA and itself and another SPA were optimal. That is, due to the design of the optical system, once the links for a SPA back to itself were aligned on all SPAs, verification of the links to one other SPA would guarantee alignment to all other SPAs. The alignment process was carried out incrementally and once completed the full test of the system was performed.

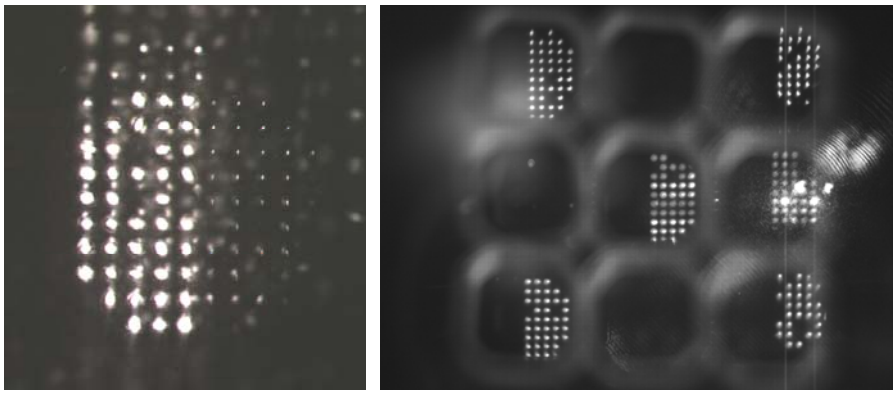

Figure 39. View of optical alignment. (left) Aligned cluster with VCSELs on the left half mapped onto detectors on the right half. (right) View of all input to one SPA. Mini-lenses for each cluster are visible.

A custom interface to the pattern generator and logic analyzer was built in order to provide manageable control over the system. The ability to independently control the four-bit bias current setting and four-bit modulation current setting for every VCSEL driver in the system as well as a digital enable for every receiver in the system and output selection bits associated with every electrical output gave remarkable flexibility to configure and test the system. It also created a need to program 3,296 registers in each ASIC, or over twenty-six thousand registers for the whole MCM. In assembling and testing an optical system the most intuitive way to visualize the optical links is by their physical location. Therefore, the interface developed for the VIVACE test bed graphically represented the physical location of each optoelectronic device and provided the appropriate controls. This was implemented with a Microsoft Excel spreadsheet using sixty-four worksheets to represent the sixty-four clusters of devices in the system. All of the mapping between physical location for a given circuit and its location within the scan chain was built into the spreadsheet. Macros within the spreadsheet provide navigation capability, utilities such as enabling or disabling all receivers within a cluster, and export the necessary patterns to be loaded into the pattern generator and then into the chips. Additional macros were added to the spreadsheet to automate the control of the pattern generator across an Ethernet connection. A sample page from this spreadsheet is shown in Figure 40.

To simplify monitoring of the bit error rate testing being performed on the optical links within the demonstration system a similar spreadsheet-based interface was created. This one controls the logic analyzer forcing it to acquire error information from the Motherboard and then import and format the data for viewing.

Since only one output out of each group of eleven can be monitored at one time, this interface arranges the results according to the physical location of the group of receivers being monitored. In addition to the number of errors, status information for the links includes whether a particular link has achieved a lock between transmitter and receiver (indicated by green highlight) and if the six-bit error counter for each link has overflowed (indicated by red highlight). A snapshot of this spreadsheet is shown in Figure 41.

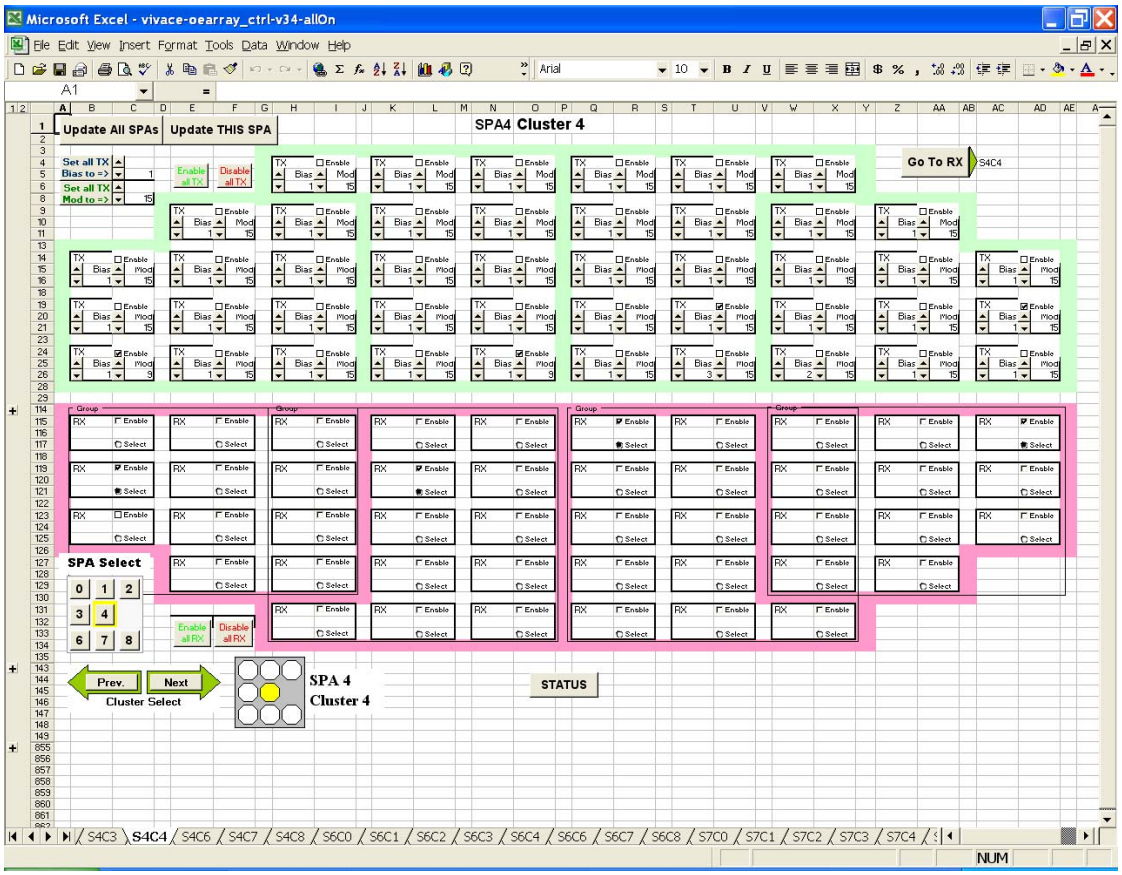

Figure 40. Sample page from test bed control spreadsheet.

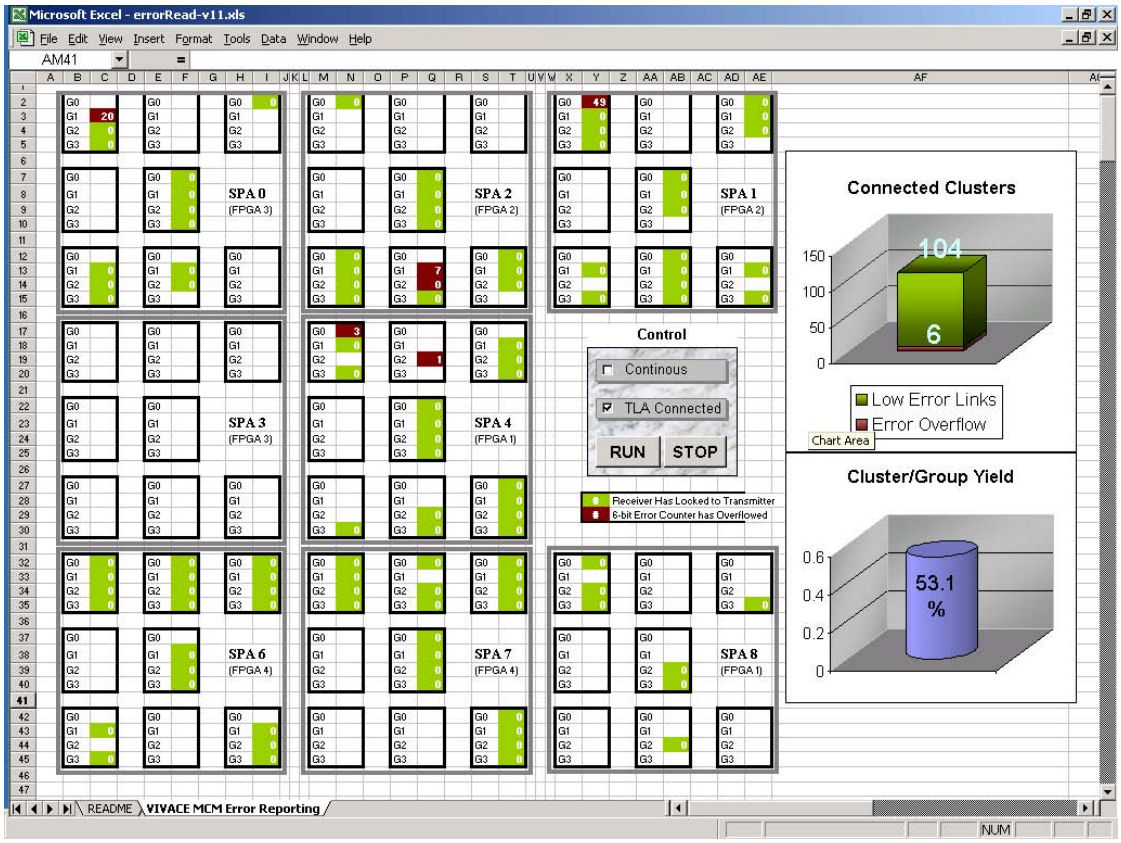

Figure 41. Snapshot of bit error checking application. Displays continuously monitored error count for all groups of links within the system.

5.3 System Test-bed Evaluation

The electronic and optoelectronic devices were demonstrated by performing functional verification and performance evaluation in stages as the final demonstration system was assembled. In addition to the testing of the CMOS die and the Motherboard assembly which has already been reported, tests were performed on the hybridized CMOS/GaAs die before being placed on the MCM and a number of tests were done with the completed MCM prior to integration with the optics and optomechanics. The electronics were exercised in the active alignment process used

to assemble the optics and further verification was carried out with the completed demonstration system.

5.3.1 Incremental Testing

Upon initial test of the Transceiver ASIC die after fabrication, a batch of sixteen silicon die was sent to Honeywell for hybridization wherein the GaAs optoelectronic device arrays were attached to them by bump-bonding. The resulting composite devices, commonly called a Smart-Pixel Arrays (SPAs), were sent back and the first optical tests were performed. Using a probe station and great care not to damage the pads (which would later be wirebonded) with the needle probes, the first demonstration of the VIVACE VCSELs being driven by the custom driver circuit was carried out. For this test, a color CCD camera with sensitivity in the near-IR region was used to visually verify the VCSEL operation. This was done by using the scan-chain interface of the Transceiver ASIC to bias the devices above their threshold. In addition to the CCD camera, an optical detector and power meter were used to verify the ability to control the light output of the VCSELs by changing the bias current setting. Absolute power measurements were not possible in this setup due to unknown optical loss in the path from the VCSEL through the micro-optics and the microscope to which the sensor was mounted, but the relative power as the bias setting was incremented was correct. The number of probes required to simultaneously make reliable contact without causing damage to the pads they were contacting made this testing very time consuming and significantly risky since the number of hybridized SPAs was limited. Therefore, this testing was only carried out for one SPA prior to assembly of the MCM. For that SPA, a VCSEL yield of 42% was observed. While this was quite low, it was very encouraging because a missing back-side processing

step in the fabrication of the GaAs wafer threatened to result in very low yield due to poor flip-chip connections between the two die.

The next testing also involved verification of the VCSEL yield after attachment and wirebonding of all eight SPAs onto the MCM. The transmitter scan chains were used to set all of the VCSEL drivers within a cluster to a bias current that should exceed the threshold current. A microscope with CCD camera was then used to image the cluster and the working and non-working VCSELs could easily be counted as shown in Figure 42. For those VCSELs that were non-functional in this test, the bias current setting was increased and the test repeated. Generally, this did not impact the yield result and it was concluded that those flip-chip bonds were open circuits as a result of the GaAs wafer-processing problem. The results of this VCSEL yield testing are summarized in Table 14. The ability to independently control the bias and modulation settings of the VCSEL drivers with this setup is exemplified in the succession of images shown in Figure 43.

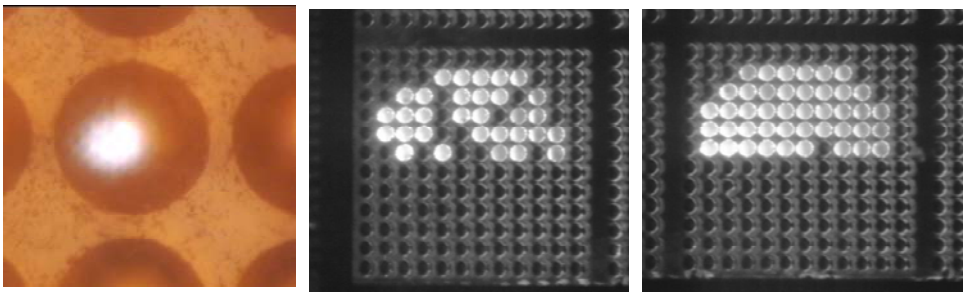

Figure 42. MCM-based test of VCSEL yield. (left) Close-up of one micro-lens with VCSEL illuminated. (center & right) View of two clusters with all functional VCSELs illuminated.

Table 14. VCSEL Yield

SPA	Yield	Percent Yield	Fault Distribution
0	283 / 352	80.4%	Fairly uniform across SPA
1	243 / 352	69.0%	Very concentrated at right side of SPA
2	259 / 352	73.3%	Uniform except for two nearly perfect clusters
3	274 / 352	77.8%	Uniform except for one nearly non-functional cluster
4	289 / 352	82.1%	Fairly uniform across SPA
6	298 / 352	84.7%	Fairly uniform across SPA
7	151 / 352	42.9%	Concentrated at lower left half of SPA
8	314 / 352	89.2%	Fairly uniform across SPA
Total	2110 / 2816	74.9%	

A number of optical power measurements were made during the course of the assembly. As discussed previously, an absolute power is difficult to measure within the system due to unknown losses, but relative power can be examined. During the initial testing of the MCM mounted onto the Motherboard, VCSEL power data was taken. This is plotted in Figure 44 for one VCSEL at different bias and modulation currents. The emission spectrum of this VCSEL was recorded as shown in Figure 45.

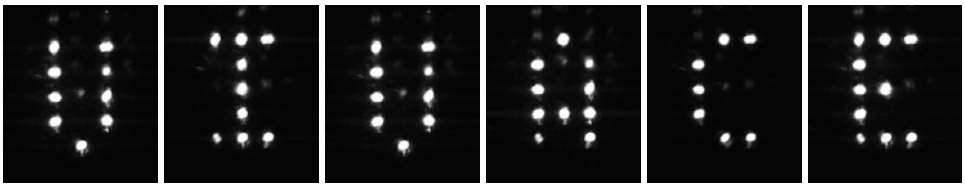

Figure 43. DC VCSEL test illustration. Independent control of VCSELs is shown in this series of photos.

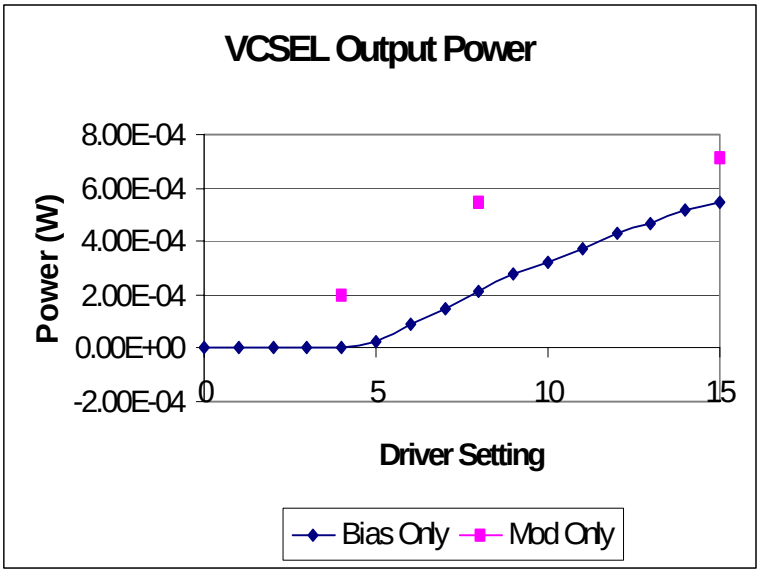

Figure 44. VCSEL light output characterization. Optical power measured versus bias and modulation setting.

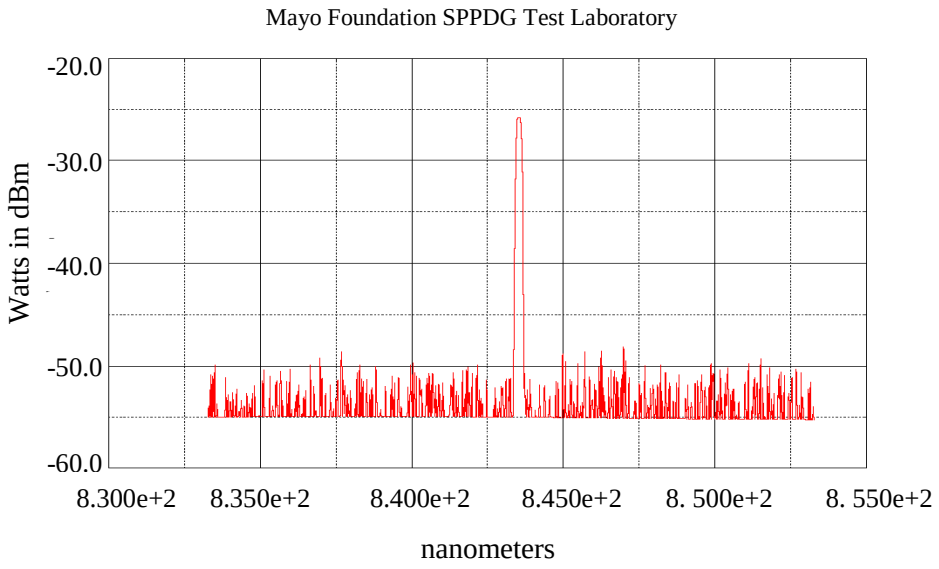

Figure 45. VCSEL light output characterization. Optical spectrum analysis of VCSEL output at threshold. Peak of –26dBm is at 843 nm. (Plot courtesy of Mayo Foundation)

Prior to beginning the optics assembly process, some initial testing of detectors and receivers was done. This allowed verification of the devices that were planned for use during the active alignment procedure. For this testing, the bare end of a fiber ribbon was positioned over the microlens of the detector to be tested using an xyz-translation stage as shown in Figure 46. The other end of the fiber was coupled to a packaged 850 nm VCSEL. The number of detectors tested in this way was very limited due to the amount of time required to exhaust the combinations of fiber alignment, optical power, and receiver gain in order to designate a detector as non-functional. Therefore, the results in Table 15 are not necessarily representative of the entire detector arrays. For comparison, the yield of a corresponding sample of VCSELs comes to roughly 80%.

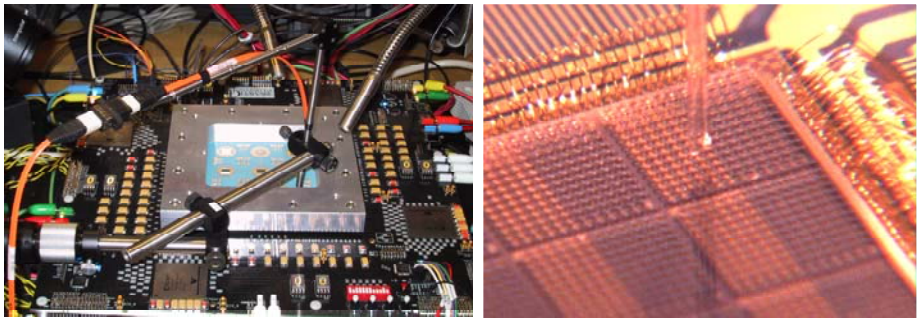

Figure 46. Fiber-based detector test. (left) Mechanical setup used to position the end of a fiber ribbon over a microlens. (right) Close-up photo of fiber in position.

Table 15. Results of Detector Testing Using Fiber-Optic Input

SPA	Yield	Percent Yield
0	18 / 20	90.0%
1	6 / 8	75.0%
2	8 / 8	100%
3	untested	
4	17 / 17	100%
6	8 / 8	100%
7	4 / 8	50.0%
8	8 / 8	100%
Total	69 / 77	89.6%

During the process of aligning the macro optics to the SPAs, several links were monitored in order to judge the quality of the alignment. For this procedure a very valuable feature of the Transceiver ASIC was used. Each of the eight clusters on the Transceiver ASIC have a VCSEL and detector pair which are designated as the "center" device and which have special functionality. For the center VCSELs, a

single digital input can be used to set the VCSEL driver input to logic high. This feature was not required in the alignment process, but the center detector functionality was very useful. For the center receiver, a dedicated output pad was allocated to allow the analog voltage from the pre-amplifier to be monitored. A large resistor at the pre-amplifier output was used to prevent this pad connection from loading the circuit and affecting its performance. The result is that these outputs can be used to determine how close to the optimum alignment a lens is for a given link instead of relying on the all-or-nothing digital output which might otherwise be used. The large series resistor does impact the speed at which the analog output can change as seen in Figure 47, but for the purposes of establishing optical alignment, speed is not critical.

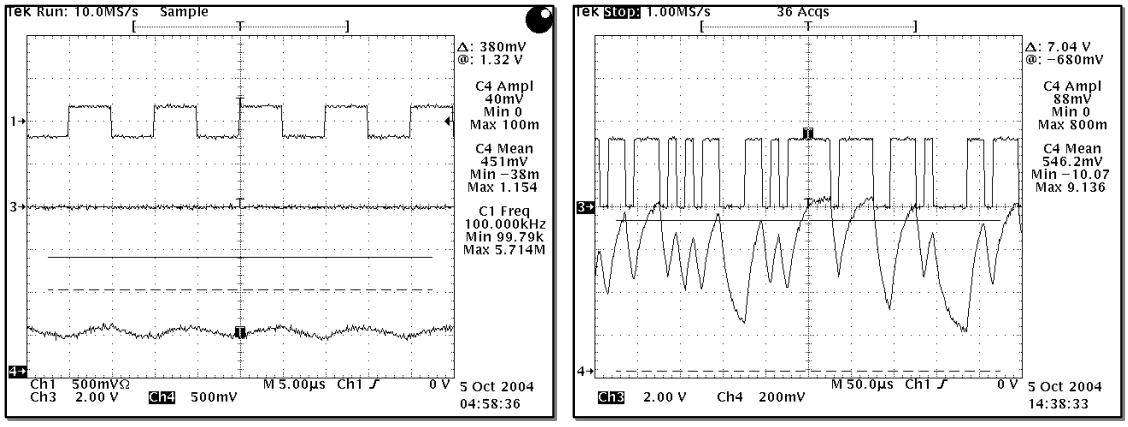

Figure 47. Analog output for active alignment. (left) Improved alignment is indicated by increased amplitude and decreased DC offset of the pre-amplifier output shown in the bottom trace. (right) Here the digital output of the receiver is shown on top and the pre-amplifier output is shown on the bottom.

5.3.2 Completed System Evaluation

Upon placing and fixing seven lenses within the demonstration system, the primary electrical evaluation task became establishing as many links as possible. The realization of the links in the completed module is illustrated in Figure 48. For this evaluation, the BERT functionality and control and monitoring applications described in the last section were used. One of the eleven links in a group can be monitored at one time, and so the output selection switches were used to determine which links were functional.

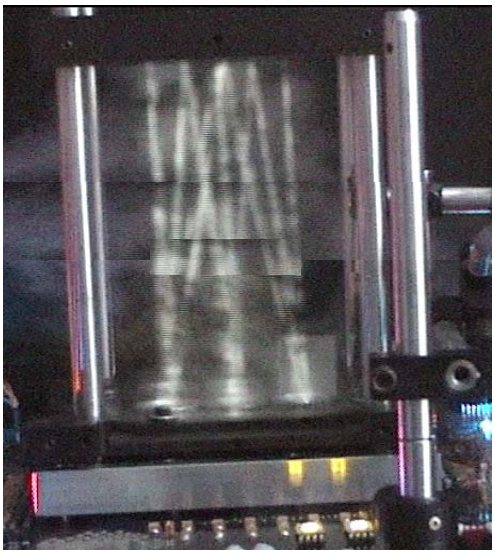

Figure 48. Light beams of aligned system. Dry-ice "fog" blown through the module allows the beams to be imaged with an IR-sensitive camera.

For a given link, many variables in addition to the device and circuit parameters determine its operation. Each VCSEL driver has 256 possible digital settings and two global analog bias settings. One of these bias settings is global for

the entire system and the other can be controlled on a per-chip basis. On the receiver side, there is one analog bias voltage for all detectors, three analog bias/reference settings that are common for the entire system, and six bias settings which can be set on a per-chip basis. The system was designed in this manner to give great flexibility to demonstrate and evaluate the optical links, but with the flexibility comes the burden of a large search space in order to find optimum operating settings. In order to make this more manageable, the number of variables was reduced by first finding settings that seemed to be best for a subset of typical links and fixing these global settings for all chips. The remainder of the link verification was carried out only changing the VCSEL bias and modulation settings. While non-functional VCSELs are fairly easy to determine, due to the test method it is overly pessimistic to classify detectors (and thus links) as non-functional because changing some of the global settings that were held fixed may have yielded a working link. This was shown to be true for some links, but due to time constraints it was not possible to independently tune and test the system for each link. Therefore, to be optimistically fair, the overall system link yield should not be calculated. Instead, the number of working links will be presented.

In the effort to enable and validate the system optical links, the primary focus was on establishing links. As such, much of the testing of the fully assembled system (shown in Figure 49) was done at low speed (2 MHz to 10 MHz). Tuning for performance or power savings was left to a later step. At this speed, more than 100 groups were demonstrated with low bit error rates. Note that each group represents eleven possible optical links between chips and one of these can be monitored for errors at a time. Increasing the data rate inversely affects the number of these groups that operate with low error rate. Working links up to 100 MHz were demonstrated.

Increasing the data rate above 100 MHz was felt to be unreliable based on the FPGA implementation, which was responsible for data generation and pattern checking. The VCSEL power of some selected links was increased and found to improve the error rate for those links. Additional tuning of system parameters, for instance optimizing the receiver gain for higher bandwidth, would likely also improve this result.

Figure 49. Photograph and close-up of the fully assembled system.

The results shown here were very encouraging despite their being less than perfect. The reason being that a single known factor was the primary cause of the reduced yield in the final system. As mentioned previously, a processing step in the fabrication of the optoelectronic device array was erroneously skipped. This resulted in highly variable reliability of the bump-bond interconnection between the Silicon and GaAs dies. Some connections were completely open, while others were marginally connected leading to much higher contact resistance. This variability (in marginal connections) led to two problems. One was that the increased resistance,

which was not designed for, pushed the interface circuits out of their operating range. The other impact was that the variability coupled with some global biasing of the interface circuits, led to links that could not be simultaneously operated at optimum or working settings. The latter problem also potentially impacts the speed that a given link can be operated at. Of course, a functional link relies on both a VCSEL/driver and photodetector/receiver operating together, and so, these effects are compounded. It is reasonable to expect that the overall system performance would be dramatically better without the bonding problem given that optoelectronic devices were tested prior to final processing (and before the missing processing step) and found to have near 100% yield.

Chapter 6

CONCLUSION

The electrical test-bed designed and built for the VIVACE demonstration was successful in its ability to exhibit the massively parallel free-space optical connectivity among multiple SPAs unique to the program. A number of challenges were overcome in order to demonstrate this hardware and some areas for improvement were observed. The large number of optical links operated in this test bed goes a long way toward demonstrating the feasibility of building more complex systems, such as the switch module design presented here, which take advantage of optical links for chip-to-chip communication.

The hardware of the VIVACE test-bed was able to perform bit-error measurements on up to 256 channels using un-correlated pseudo-random data. This capability was demonstrated in the verified operation of 110 out of a possible 196 groups of optical links. Grouping of optical links was required due to the 11-to-1 ratio by which the optical I/O bandwidth exceeded that available on the electrical ports of the SPAs. The total number of possible groups that can be linked is decreased based on demonstrating chip-to-chip connectivity with only seven of the eight SPAs on the MCM.

Some changes to the demonstrated hardware could have been made. Architectural changes would have been made to the Transceiver ASIC in order to optimize it for speed had it not been designed as a backup to the Switch ASIC. Similarly, on the Motherboard, the architecture was targeted for the switch design and

FPGA devices used would have been different had the original program goal been high-speed operation of optical links. In support of this claim, it is noted that in another program shared by many of the VIVACE team members, narrow optical channels, which used VCSELs and photodetectors and similar driver and receiver circuits, were demonstrated at 2.5 Gbps.

While the links were demonstrated at relatively low speed, the goal of demonstrating parallel free-space optical links was achieved. There is also great promise to extending the operating speed based on inherent bottlenecks in the current system that could be removed in a future system design.

Extending the approach taken in the VIVACE program can be done in a number of ways. The use of free-space communication links between silicon CMOS ICs increases the complexity, cost, and risk of the system design; therefore, justification for the inclusion of such communication must be considered. For the switch design presented here, the motivation to use optical interconnect is centered on scalability. While it is true that switches can be built and scaled without using optical interconnect, a fundamental problem is encountered in the scaling process. As the number of chips within a switch is increased, it becomes impossible to maintain full connectivity between the chips due to the limited I/O bandwidth of each chip and the difficulty in creating the all-to-all routing in an electrical substrate. This type of connectivity is inherently difficult to implement electrically. However, The limited interference of adjacent free-space optical links makes them ideal for such dense global interconnect patterns. In practice, non-blocking or limited blocking properties are maintained when scaling a switch design by using multi-stage interconnection networks to create larger switches. However, such scaling methods require larger

systems. Nonetheless, such scaling strategies can be applied to the case of an optically interconnected switch fabric as well, with the added benefit of using larger non-blocking sub-components.

Further scaling strategies are also possible. These include standard evolutionary changes as well as architectural modifications. Modifying the electrical and optical interfaces to use double data-rate (DDR) signaling would allow an immediate doubling of the number of ports by using two switch cores per ASIC to handle data for two switch ports. Optical DDR links have been demonstrated in [64]. Additionally, newer CMOS fabrication technologies would allow the internal logic data rates to be increased as seen in other commercial ICs over time. It should be pointed out that transmitter and receiver circuitry as well as similar optoelectronic devices have been demonstrated at much higher frequencies than what has been used here. While the goal of this design was to implement wide optical interconnects that run at the internal logic data rate, similar demonstrations of a few optical links running at several gigabits per second each have been demonstrated [65]. Using high-speed serial links, which are becoming more prevalent in digital systems, to connect directly to the switch fabric and potentially in the optical links within the switch could dramatically increase the number of ports handled by a single switch ASIC. Thus, architectural changes coupled with future research and newer fabrication processes could lead to much greater numbers of ports and port bandwidth.

Physically scaling the design in some ways may also be possible. Scaling the number of fabric chips on a single MCM is difficult due to size restrictions in the technology that is used to fabricate it and the physical size of the optics. However, there is room within the existing optical system design to use larger optoelectronic

device arrays with the same die-to-die pitch. Based on the favorable test results achieved in this test system, other substrate materials could be considered for future systems, which could greatly increase the size of the switch fabric possible. Finally, scaling in the traditional sense – by using multiple switches – is possible with the current and future designs.

The hardware demonstrated and presented here illustrates the feasibility of a new technology that can be used to build improved digital systems. As discussed, further evaluation of the VIVACE demonstration module is possible, but it is felt that the experience gained and data collected from this system would be immediately useful in continuing this research and extending it to even larger systems of free-space optical interconnect.

BIBLIOGRAPHY

[1] Michael W. Haney, Marc P. Christensen, Predrag Milojkovic, Gregg J. Fokken, Mark Vickberg, Barry K. Gilbert, James Rieve, Jeremy Ekman, Premanand Chandramani, Fouad Kiamilev, "Description and Evaluation of the FAST-Net Smart Pixel-Based Optical Interconnection Prototype," *Proceedings of the IEEE*, vol. 88, No. 6, pages 819-828, June 2000.

[2] Simon Stanley, "Traffic Manager Chips", *Light Reading*, [Online], October 17, 2002, Available: http://www.lightreading.com/document.asp?site=lightreading&doc_id=22628 .

[3] "CSIX-L1: Common Switch Interface Specification-L1," *Network Processing Forum Implementation Agreement*, August 5, 2000.

[4] "System Packet Interface Level 4 (SPI-4) Phase 2: OC-192 System Interface for Physical and Link Layer Devices," *Optical Internetworking Forum Implementation Agreement*, January 2001.

[5] Simon Stanley, "Switch-Fabric Chipsets", *Light Reading*, [Online], March 3, 2004, Available: http://www.lightreading.com/document.asp?doc_id=47959.

[6] David M. Ewalt, "Sprint Makes Big Bet On Packet Technology," *Information Week* June 2, 2003.

[7] "Deployment of Wireline Services Offering Advanced Telecommunications Capability," FCC 99-48, released March 31, 1999. Available: http://www.fcc.gov/Bureaus/Common_Carrier/Orders/1999/fcc99048.pdf.

[8] Federal Standard 1037C: Glossary of Telecommunications Terms Website, Available: http://www.its.bldrdoc.gov/fs-1037/.

[9] Harry Newton, *Newton's telecom dictionary: the official dictionary of telecommunications networking*, New York: CMP Books, 2000.

[10] Lawrence Roberts, "The evolution of packet switching," *Proceedings of the IEEE*, vol. 66, no. 1, November 1978.

[11] Hein, *Switching Technology in the Local Network*, London: International Thompson Computer Press, 1997.

[12] D-link Website, Available: http://dlink.com/.

[13] Netgear Website, Available: http://netgear.com/.

[14] P. Kermani and L. Kleinrock, "Virtual cut-through: A new computer communication switching technique," *Computer Networks*, vol. 3, 1979, pp. 267-286.

[15] W.J. Dally and C.L. Seitz, "The Torus Routing Chip," *J. of Distributed Computing*, vol. 1, no. 3. 1986, pp. 187-196.

[16] J.T. Draper and J. Gosh, "A comprehensive analytical model for wormhole routing in multicomputer systems," *J. Parallel and Distributed Computing*, vol. 23, no. 2, pp. 202-214, Nov. 1994.

[17] Pierre Guerrier, "A Generic Architecture for On-Chip Packet-Switched Interconnections," in *Proceedings of DATE 2000* (Paris France, March 27-30, 2000).

[18] Dally, W. J., Fiske, J. S., Keen, J. S., Lethin, R. A., Noakes, M. D., Nuth, P. R., Davison, R. E., And Fyler, G. A., "The Message-Driven Processor: A multicomputer processing node with efficient mechanisms", *IEEE Micro*, pp. 23–39, Apr. 1992.

[19] D. Lenoski, J. Laudon, K. Gharachorloo, W.-D. Weber, A. Gupta, J. Hennessy, M. Horowitz, and M. S. Lam, "The Stanford Dash Multiprocessor," *IEEE Computer*, pp. 63-79, Mar. 1992.

[20] Kessler, R.E. And Schwarzmeier, J. L. "CRAY T3D: A new dimension for Cray research", in *Proceedings of IEEE CompCon*, (Spring 1993, pp. 176–182).

[21] Leiserson, C. E., Abuhamdeh, Z. S., Douglas, D. C., Feynman, C. R., Ganmukhi, M. N., Hill, J. V., Kuszmaul, B. C., Pierre, M. A. S., Wells, D. S., Wong, M. C., Yang, S. W., and Zak, R., "The network architecture of the connection machine CM-5", in *Proceedings of the ACM Symposium on Parallel Algorithms and Architectures*, (1992, pp. 544–557).

[22] Abali, B. and Aykanat, C., "Routing algorithms for IBM SP1", in *Proceedings of the Parallel Computer Routing and Communications Workshop*, (May 1994, pp. 161–175).

[23] Prasant Mohapatra, "Wormhole Routing Techniques for Directly Connected Multicomputer Systems," *ACM Computing Surveys*, vol. 30, no. 3, September 1998.

[24] Ni, "A Survey of Wormhole Routing Techniques in Direct Networks", *IEEE Computer*, Feb. 1993.

[25] Kumar, "The Sliding-Window Packet Switch: A New Class of Packet Switch Architecture With Plural Memory Modules and Decentralized Control," *IEEE J. Sel. Areas in Communications*, May 2003.

[26] Simon Stanley, "Packet Switch Chips", *Light Reading*, [Online], Feb. 2, 2003, Available: http://www.lightreading.com/document.asp?doc_id=25989.

[27] D. K. Hunter, "Switching Systems", *Encyclopedia of Information Technology*, vol. 42, supplement 27, A. Kent, J. G. Williams, C. M. Hall (Eds.), Marcel Dekker, New York: Basel, 2000, pp. 335-370.

[28] Y. Yeh, M. G. Hluchyj, and A. S. Acampora, "The Knockout Switch: A Simple, Modular Architecture for High-Performance Packet Switching," *IEEE J. Sel. Areas Commun. 5*, no. 8, 1274 –1283, October 1987.

[29] Mark J. Karol, Michael G. Hluchyj and Samuel P. Morgan "Input Versus Output Queueing on a Space-Division Packet Switch," *IEEE Transactions on Communications*, vol. 35, no. 12, pp. 1347 – 1356, December 1987.

[30] Y. Tamir and G. Frazier, "High performance multi-queue buffers for VLSI communication switches", in *Proceedings of 15th Annual Symp. Computer Arch.*, June 1988, pp. 343-354.

[31] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 2, pp. 188-201, April 1999.

[32] C.Minkenberg, T.Engbersen, "A Combined Input and Output Queued Packet-Switched System Based on PRIZMA Switch-on-a-Chip Technology," *IEEE Communications Magazine*, pp. 70-77, Dec. 2000.

[33] S.T. Chuang, A. Goel, N. McKeown and B. Prabhkar, "Matching output queueing with a combined input output queued switch," in *Proceedings of IEEE INFOCOM '99*, pp. 1169-1178, (New York, 1999).

[34 Manolis Katevenis, Georgios Passas, Dimitrios Simos, Ioannis Papaefstathiou, and Nikos Chrysos, "Variable Packet Size Buffered Crossbar (CICQ) Switches," in *Proceedings of the IEEE Int. Conference on Communications (ICC 2004)*, (Paris, France, 20-24 June 2004).

[35] Myricom website, Available: http://www.myri.com/myrinet/overview/index.html.

[36] Z. Haas, D.R. Cheriton, "Blazenet: a packet-switched wide-area network with photonic data path," *IEEE Transactions on Communications*, Vol. 38, Issue: 6, pp: 818-829, June 1990.

[37] Duato, J., Yalamanchili, S., Ni, L., *Interconnection Networks: An Engineering Approach*, Los Alamitos, CA: IEEE Press, 1997.

[38] D. N. Serpanos and P. I. Antoniadis, "Firm: A Class of Distributed Scheduling Algorithms for High-Speed ATM Switches with Multiple Input Queues," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2000)*, (Tel Aviv, Israel, March 2000, vol. 2, pp. 548-555).

[39] Jajszczyk, "Nonblocking, Repackable, and Rearrangeable Clos Networks: Fifty Years of the Theory Evolution," *IEEE Communications Magazine*, 2003.

[40] Chao, "Matching Algorithms for Three-Stage Bufferless Clos Network Switches," *IEEE Communications Magazine*, vol.41, no.10, 2003.

[41] "Device Specification: S2065 Quad serial backplane with Dual I/O," Applied Micro Circuits Corporation, May 2000, Available: http://www.amcc.com.

[42] "IEEE standard test access port and boundary-scan architecture," *IEEE Std 1149.1-2001*, 2001, pp. i-200.

[43] Carl Wilmsen, Henryk Temkin, Larry Coldren, eds. *Vertical-Cavity Surface-Emiting Lasers: Design, Fabrication, Characteization and Applications*, Cambridge, UK: Cambridge University Press, 1999.

[44] Towe *et al.* "A historical perspective of the development of the vertical-cavity surface-emitting laser," *IEEE Journal on Selected Topics in Quantum Electronics*, vol. 6, no. 6, pages: 1458-1464, November/December 2000.

[45] Liu, "Heterogeneous integration of OE arrays," *IEEE Transactions on Advanced Packaging*, vol. 25, no. 1, pages 43-49, February 2002.

[46] James K. Guenter, Jim A. Tatum, Andrew Clark, R. Scott Penner, Ralph H. Johnson, Robert A. Hawthorne, J. Robert Biard, and Yue Liu, "Commercialization of Honeywell's VCSEL Technology: Further Developments," *Proceedings of the SPIE,* vol. 4286, 2001.

[47] S.D. Personick, "Receiver design for digital fiber-optic communication systems," *Bell Syst. Tech. J.*, 52(6), pp. 843-886, July 1973.

[48] A. Buchwald and K. Martin. *Integrated fiber optics receivers*, Boston: Kluwer Academic Publishers, 1995.

[49] T.K. Woodward, A.V. Krishnamoorthy, A.L. Lentine, and L.M.F. Chirovsky. "Optical receivers for optoelectronic VLSI," *IEEE J. on Sel. Top. in Quant. Elec.*, vol. 2, no. 1, April 1996.

[50] F.E. Kiamilev and A.V. Krishnamoorthy. "A 500Mb/s, 32- channel CMOS VCSEL Driver with built-in self-test and clock generation circuitry," Submitted to *IEEE Journal of Quantum Electronics*, Nov. 1998.

[51] A.V. Krishnamoorthy. "3-D integration of MQW modulators over active sub-micron CMOS circuits: 375mb/s transimpedance receiver-transmitter circuit," *IEEE Photonics Technology Letters*, vol.7, no. 11, pp. 1278-1290, Nov. 1995.

[52] L.F. Miller. "Controlled Collapse Reflow Chip Joining," *IBM Journal of Research and Development*, vol. 13, no. 3, pp. 239-250, 1969.

[53] Richard Rozier, Ray Farbarik, Fouad Kiamilev, Jeremy Ekman, Premanand Chandramani, Ashok V. Krishnamoorthy, and Richard Oettel. “Automated Design of Integrated Circuits with Area-Distributed Input-Output Pads,” *Applied Optics*, 10 September 1998.

[54] Jeremy Ekman, Fouad Kiamilev, Gregg Fokken, Scott Sommerfeldt, Barry Gilbert, Mark Vickberg, Yue Liu, Allen Cox, Ping Gui, Premanand Chandramani, Xiaoqing Wang, Michael Haney, Marc Christensen, Predrag Milojkovic, Kevin Driscoll, Brian Vanvoorst, "System design and packaging for an optically interconnected mcm switch for parallel computing", in *Proceedings of Interpack 2001*, (Kauai, Hawaii, USA, July 8-13, 2001).

[55] Turner. "Multirate Clos Networks,” *IEEE Communications Magazine*, October 2003.

[56] “TSMC 0.25-micron Technology Platform”, *Taiwan Semiconductor Manufacturing Company Limited* Available: http://www.tsmc.com.

[57] P. Chandramani, P. Gui, J. Ekman, X. Wang, F. Kiamilev, M. Christensen, P. Milojkovic, M. Haney, J. Anderson, K. Driscoll, B. Vanvoorst, "Design of a Multi-Gigabit Optical Network Interface Card," *Selected Topics in Quantum Electronics, IEEE Journal on*, vol. 9, iss. 2, pp. 636-646, March-April 2003.

[58] F. Kiamilev, P. Chandramani, P. Gui, J. Ekman, B. Vanvoorst, F. Rose, K. Driscoll, J.A. Cox, M. Christensen, P. Milojkovic, M. Haney, "Programmable Network Interface for Parallel Optical Data Links," in *Proc. CLEO/Europe'00 Conf.* (Nice, France, September 2000).

[59] P. Gui, P. Chandramani, J. Ekman, X. Wang, F. Kiamilev, K. Driscoll, B. Vanvoorst, Y. Liu, J. Nohava, J. A. Cox, M. Christensen, M. Haney, and P. Milojkovic, "Gigabit Optical Network Interface Card using Parallel Data Fiber Link for a Free-Space Switched Local Area Network System," in *Proc. IEEE LEOS 2001 Annual Meeting*, (San Diego, CA, November 2001, pp. 863-864).

[60] Y. Liu, "Heterogeneous integration of OE arrays with Si electronics and microoptics," *IEEE Transactions on Advanced Packaging*, vol. 25, no.1, pp.43-49, Feb. 2002.

[61] Microfab Technologies Inc. Corporate website. Available: www.microfab.com.

[62] P. Milojkovic, M.P. Christensen, M.W. Haney, **"**Multi-scale lens design for the global multi-chip FAST-Net interconnection module," in *Proc. The 14th Annual Meeting of the IEEE Lasers and Electro-Optics Society* (Nov. 2001, vol.2, pp. 814 – 815).

[63] M.P. Christensen, P. Milojkovic, M.J. McFadden, M.W. Haney, "Multiscale optical design for global chip-to-chip optical interconnections and misalignment tolerant packaging," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 9, iss. 2, pp. 548 – 556, March-April 2003.

[64] Ping Gui, Fouad Kiamilev, Xiaoqing Wang, Michael McFadden, Jeremy Ekman, Joseph Deroba, Charlie Kuznia, Michael Haney, "Source-synchronous Double Data Rate (DDR) parallel optical interconnects," in *Proceedings of InterPACK03, the PACIFIC RIM/International, Intersociety, Electronic Packaging Technical/Business Conference & Exhibition*, Maui, Hawaii, USA, July 6-11, 2003.

[65] X. Wang, F. Kiamilev, P. Gui, J. Ekman, G. C. Papen, M. J. McFadden, M. W. Haney, C. Kuznia, "A 2-Gb/s optical transceiver with accelerated bit-error-ratio test capability," *Journal of Lightwave Technology*, vol. 22, iss. 9, pp. 2158 – 2167, Sept. 2004.